



ALAT PEMBERI MAKAN IKAN DI AKUARIUM
OTOMATIS BERBASIS MIKROKONTROLER ATMEGA16
PROYEK AKHIR

Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta
Untuk Memenuhi Sebagian Persyaratan
Guna Memperoleh Gelar Ahli Madya Teknik



Oleh
Lukman Nulhakim
NIM. 09506131021

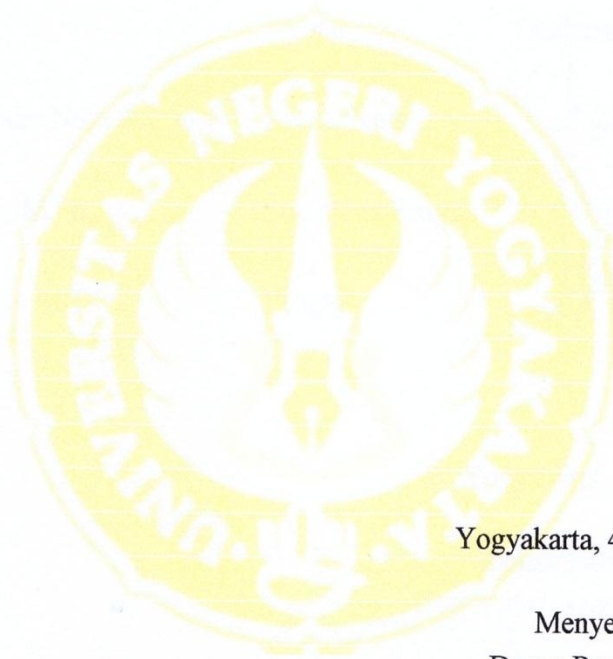
PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA

2014

HALAMAN PERSETUJUAN

PROYEK AKHIR

Proyek Akhir yang berjudul “Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroller ATmega 16” ini telah disetujui oleh pembimbing untuk diujikan.



Yogyakarta, 4 April 2014

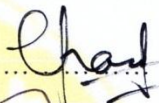

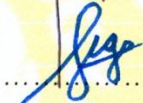
Menyetujui,
Dosen Pembimbing

Ariadie Chandra Nugraha, M.T.
NIP. 19770913 200501 1 002

PENGESAHAN

Proyek akhir yang berjudul “Alat Pemberi Makan Ikan Di Akuarium Otomatis Berbasis Mikrokontroller ATmega 16” ini telah dipertahankan di depan Dewan Penguji pada tanggal 14 April 2014 dan dinyatakan lulus.


SUSUNAN TIM PENGUJI

Nama	Jabatan	Tandatangan	Tanggal
Ariadie Chandra Nugraha, M.T.	Ketua Penguji		22/4 2014
Toto Sukisno, M.Pd.	Sekretaris Penguji		24/4 2014
Sigit Yatmono, M.T.	Penguji		24/4 2014

Yogyakarta, 22 April 2014

Dekan Fakultas Teknik
Universitas Negeri Yogyakarta,




Dr. Moch. Bruri Triyono

NIP. 19560216 198603 1 003

LEMBAR PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa dalam Proyek Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya atau gelar lainnya di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali secara tertulis yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Pembuatan Proyek Akhir ini dibawah bimbingan Ariadie Chandra Nugraha, M.T.

Yang menyatakan,



Lukman Nulhakim

NIM : 09506131021

MOTTO

“Berjuang untuk perbaikan”

“Sebaik-baik manusia adalah yang bermanfaat untuk orang lain”

“Kesuksesan adalah ketika kita bermanfaat bagi orang lain”

PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT, karya ini ku persembahkan kepada :

- ❖ Ayah, Ibu dan kakakku tercinta yang telah memberikan dorongan dan kasih sayangnya serta selalu mendoakanku sepanjang waktu untuk keberhasilanku.
- ❖ Untuk keluarga besarku yang selalu memberi dukungannya.
- ❖ Segenap Dosen UNY terutama Dosen Elektro yang selalu memberi solusi, dukungan, dan motivasi.
- ❖ Kawan-kawanku seperjuangan Kelas B Teknik Elektro kelas B 2009 yang telah memberikan hangatnya kekeluargaan selama dibangku kuliah.
- ❖ Kawan-kawan Penegak Kedisiplinan MAKRAB dan OSPEK UNY
- ❖ Kawan-kawan Masjid Al-Falaah yang telah memberikan pelajaran dan pengalaman yang sangat berharga serta kekeluargaan yang terjalin.

Terima kasih atas segala doa, bimbingan, kasih sayang, dan pengorbanan yang tidak pernah berhenti.

“ALAT PEMBERI MAKAN IKAN DI AKUARIUM OTOMATIS BERBASIS MIKROKONTROLLER ATMEGA 16”

Oleh :

**Lukman Nulhakim
NIM. 09506131021**

ABSTRAK

Tujuan pembuatan alat yang berjudul “Alat Pemberi Makan Ikan Di Akuarium Otomatis Berbasis Mikrokontroler ATmega 16” adalah untuk mempermudah dalam perawatan ikan di akuarium terutama apada pemberian pakannya, sehingga ketika pemelihara ikan memiliki kesibukan atau mendapatkan kendala ketika meninggalkan ikan di akuariumnya dalam jangka waktu lama, ikan akan tetap terjaga dalam proses pemberian pakannya.

Sistem yang dirancang terdiri dari empat bagian yaitu: catu daya, sistem *minimum*, rangkaian driver dan program. Catu daya merupakan sumber daya untuk menjalankan seluruh sistem yang terdiri dari tegangan AC yang disearahkan menjadi DC oleh *rectifier*. Sistem *minimum* berupa rangkaian elektronik yang didesain sedemikian rupa sehingga dapat berfungsi sebagai pengolah data dengan mikrokontroler ATmega16 sebagai pusat kendali. Bagian ketiga adalah rangkaian driver yang berfungsi untuk mengatur buka tutup solenoid pada alat pemberi makan ikan. Bagian yang keempat adalah program yang berfungsi untuk mengatur mikrokontroler sehingga dapat bekerja sesuai dengan fitur yang ditawarkan.

Berdasarkan hasil pengujian dan untuk kerja dari “Alat Pemberi Makan Ikan Di Akuarium Otomatis Berbasis Mikrokontroler ATmega 16” telah menunjukkan hasil yang sesuai dengan perencanaan yaitu alat dapat memberikan pakan ikan secara otomatis pada waktu yang telah ditentukan. Rata-rata berat pakan yang dikeluarkan yaitu sebesar 1,6 gram pada waktu buka solenoid 500 ms, 4,82 gram pada waktu buka solenoid 1000 ms dan 8,35 gram pada waktu buka solenoid 1500 ms .

Kata kunci : Akuarium, Pakan Ikan, Mikrokontroler ATmega16

KATA PENGANTAR



Puji syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat, karunia dan nikmat yang telah diberikan-Nya, sehingga Penulis dapat menyelesaikan tugas akhir dan penyusunan laporan ini.

Penulis sadar tanpa bantuan berbagai pihak Tugas Akhir ini tidak akan terselesaikan dengan baik. Oleh karena itu pada kesempatan ini penulis dengan segala kekurangan mengucapkan terima kasih atas dukungan, bimbingan dan bantuannya baik secara moril maupun materiil kepada :

1. Bapak Ariadie Chandra Nugraha, M.T. selaku Penasehat Akademik sekaligus Pembimbing Proyek Akhir.
2. Bapak Sigit Yatmono, M. T. selaku Penguji Proyek Akhir.
3. Bapak Toto Sukisno, M.Pd. selaku Sekretaris Penguji dan Koordinator Proyek Akhir.
4. Para Teknisi Laboratorium dan Bengkel Elektro atas bantuan peralatannya.
5. Teman-teman Kelas B angkatan 2009 yang telah memberikan banyak semangat.
6. Semua pihak yang telah banyak memberikan bantuan baik moril maupun materiil untuk terselesaikannya proyek akhir ini.

Penulis menyadari bahwa Laporan Tugas Akhir ini masih banyak kekurangannya, oleh karena itu saran dan kritik yang membangun sangat penulis harapkan demi kesempurnaan di masa yang akan datang. Akhirnya Penulis

berharap semoga Laporan Tugas Akhir ini dapat bermanfaat bagi diri Penulis dan pembaca semuanya.

Yogyakarta, April 2014

Lukman Nulhakim
NIM. 09506131021

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN	iv
MOTTO.....	v
PERSEMBAHAN.....	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Identifikasi Masalah	2
C. Batasan Masalah.....	3
D. Rumusan Masalah	3
E. Tujuan.....	3
F. Manfaat.....	4
G. Keaslian Gagasan	5

BAB II PENDEKATAN PEMECAHAN MASALAH

A. Catu Daya	6
B. Trafo Step Down.....	8
C. Driver Solenoid	11
1. <i>Relay</i>	11
2. Transistor.....	12
3. Resistor.....	14
4. LED (<i>light-emitting diode</i>).....	14
D. Solenoid	16
E. <i>Liquid Crystal Display (LCD)</i>	17
F. Push Button.....	18
G. <i>Keypad</i>	19
H. Sensor Objek Benda	19
1. Photo Dioda.....	20
2. LED.....	21
I. Mikrokontroler Atmega16.....	21
1. Arsitektur ATmega16	22
2. Konfigurasi Pin Atmega16.....	23
J. Diagram Alir (<i>Flowchart</i>).....	26

BAB III KONSEP PERANCANGAN

A. Analisis Kebutuhan	28
B. Perancangan Perangkat Keras.....	30
1. Perancangan Rancangan Catu Daya	30

2. Perancangan Rancangan Keypad 3X4	31
3. Rangkaian Sistem Minimum Mikrokontroler Atmega16	32
4. Rangkaian LCD	33
5. Rangkaian Driver Splenoid	33
6. Pembuatan <i>Printed Circuit Board (PCB)</i>	34
7. Pembuatan Box.....	36
C. Perancangan Perangkat Lunak	37
D. Perancangan Pengujian Pengambilan Data	39
1. Langkah-langkah Pengambilan Data	39
2. Alat dan Bahan yang Digunakan	39
3. Perancangan Tabel Pengujian	40

BAB IV HASIL PENGUJIAN DAN PEMBAHASAN

A. Hasil Pengujian Alat.....	44
B. Pembahasan Hasil Pengujian	47
1. Catu Daya (<i>Power Supply</i>).....	47
2. Pengamatan Tampilan LCD	48
3. Pengamatan Fungsional Alat.....	49
a. Tampilan LCD	49
b. <i>Keypad</i>	53
c. Tombol Menu	55
d. Tombol Manual	56
e. Tombol Semi-Auto.....	56
f. Tombol Otomatis.....	57

g. Solenoid	58
h. Led Indikator.....	59
4. Pengamatan Ketepatan Waktu Alat	60
5. Pengamatan Berat Pakan Yang Dikeluarkan.....	61
a. Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 500ms.....	61
b. Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 1000ms.....	62
c. Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 1500ms.....	63
 BAB V KESIMPULAN DAN SARAN	
A. Kesimpulan	65
B. Keterbatasan Alat	66
C. Saran	66
DAFTAR PUSTAKA	67
LAMPIRAN	69

DAFTAR TABEL

	Halaman
Tabel 1. Pin Description: 78xx	7
Tabel 2. Konfigurasi Pin LCD.....	17
Tabel 3. Fungsi Khusus masing-masing <i>pin</i> pada <i>PORT B</i>	24
Tabel 4. Fungsi Khusus masing-masing <i>pin</i> pada <i>PORT C</i>	25
Tabel 5. Fungsi Khusus masing-masing <i>pin</i> pada <i>PORT D</i>	25
Tabel 6. Simbol-simbol dalam diagram alir.....	27
Tabel 7. Pengujian Rangkaian Catu Daya.....	40
Tabel 8. Pengujian Tampilan LCD.....	41
Tabel 9. Pengujian Fungsional Alat.....	41
Tabel 10. Pengujian Ketepatan Waktu Alat	42
Tabel 11. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 500 ms	42
Tabel 12. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 1000 ms.....	42
Tabel 13. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 1500 ms.....	43
Tabel 14. Pengujian Rangkaian Catu Daya.....	45
Tabel 15. Pengujian Tampilan LCD.....	45
Tabel 16. Pengujian Fungsional Alat.....	46
Tabel 17. Pengujian Ketepatan Waktu Alat	46

Tabel 18. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu Buka	
Solenoid 500 ms	46
Tabel 19. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu Buka	
Solenoid 1000 ms.....	47
Tabel 20. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu Buka	
Solenoid 1500 ms.....	47

DAFTAR GAMBAR

	Halaman
Gambar 1. Konfigurasi Pin pada LM 78xx	7
Gambar 2. Rangkaian Catu Daya 5 Volt.....	8
Gambar 3. Gambar Transformator <i>Stepdown</i>	10
Gambar 4. Simbol Transformator <i>Step Down</i>	11
Gambar 5. Simbol diagram <i>relay</i>	12
Gambar 6. Transistor	13
Gambar 7. Resistor	14
Gambar 8. Resistor	15
Gambar 9. Bagian-bagian Solenoid	16
Gambar 10. Solenoid	16
Gambar 11. Konfigurasi Pin LCD Karakter 16x2.....	17
Gambar 12. Push Button	18
Gambar 13. Gambar <i>Keypad 3x4</i>	19
Gambar 14. Rangkaian Sensor Benda	20
Gambar 15. Konfigurasi Pin ATmega-16	23
Gambar 16. Diagram Blok Rangkaian.....	29
Gambar 17. Rangkaian Catu Daya	31
Gambar 18. Rangkaian keypad 3X4	31
Gambar 19. Skema Rangkaian Sistem Minimum Mikrokontroler Atmega16.	32
Gambar 20. Skema Rangkaian LCD Monitor 16x2	33
Gambar 21. Rangkaian Driver Solenoid	34

Gambar 22. Rangkaian Driver Solenoid	35
Gambar 23. Sketsa Rancang Bangun Box Alat.....	36
Gambar 24. Box Alat tampak keseluruhan	37
Gambar 25. Diagram Alir Program Utama	38
Gambar 26. Tampilan Untuk Waktu Buka Solenoid.....	49
Gambar 27. Tampilan Pada Saat Countdown	50
Gambar 28. Tampilan Untuk Tombol Menu.....	55
Gambar 29. Tampilan Untuk Tombol Semi-Auto.....	56
Gambar 30. Grafik Berat Pakan Yang Dikeluarkan	64

DAFTAR LAMPIRAN

Lampiran 1. Listing Program Lengkap

Lampiran 2. Data Sheet Atmega16

Lampiran 3. Datasheet BD135/137/139

Lampiran 4. Datasheet Pushbutton

Lampiran 5. Datasheet Resistor

Lampiran 6. Rangkaian Keseluruhan

Lampiran 7. Foto Alat

BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi di bidang elektronika dewasa ini berkembang sangat pesat dan berpengaruh dalam pembuatan alat-alat yang canggih, yaitu alat yang dapat bekerja secara otomatis dan memiliki ketelitian tinggi sehingga dapat mempermudah pekerjaan yang dilakukan oleh manusia menjadi lebih praktis, ekonomis dan efisien. Perkembangan teknologi tersebut telah mendorong kehidupan manusia untuk hal-hal yang otomatis. Otomatisasi dalam semua sektor yang tidak dapat dihindari, sehingga penggunaan yang awalnya manual bergeser ke otomatisasi. Tidak terkecuali dengan hobi seperti memelihara ikan dalam akuarium yang dapat menggunakan alat sebagai pembantu untuk kemudahan dalam penggunaannya.

Dalam kehidupan sehari-hari baik itu di kota ataupun di pedesaan, terdapat banyak pemelihara ikan dalam akuarium baik yang berukuran besar, sedang maupun yang berukuran kecil. Memelihara ikan adalah suatu hobi masyarakat yang sangat digemari dari dulu hingga sekarang, karena kemudahannya dalam pemeliharaan dan perawatannya yang membuat kebanyakan orang ingin memelihara ikan. Ikan yang dipelihara dalam akuarium harus diperhatikan waktu pemberian pakannya sehingga ikan tersebut membutuhkan jadwal pemberian pakan yang teratur dan terus menerus. Namun karena kesibukan atau kegiatan lain dan di luar dugaan,

seringkali menjadi kendala pada saat pemberian pakan pada ikan di akuarium tersebut. Kendala ketika seseorang harus berpergian jauh hingga memakan waktu yang lama sampai berhari-hari, pasti akan berpikir bagaimana dengan keadaan ikan-ikan yang dipelihara dan bagaimana cara agar bisa memberi makan ikan-ikan tersebut dengan terus menerus atau terjadwal tanpa harus mengganggu aktivitas sehari-hari.

Dari permasalahan tersebut maka dibutuhkan suatu alat yang dapat memberi makan ikan secara otomatis, yang mampu melakukan pemberian pakan ikan secara otomatis pada waktu-waktu yang telah ditentukan yaitu dengan mengatur waktu pemberian pakan sesuai dengan jadwal yang diinginkan pengguna. Dengan pemberian pakan yang sudah dirancang secara otomatis pengguna tersebut tidak perlu khawatir lupa atau harus ada pada saat memberi makan ikan peliharaannya. Berdasarkan dari latar belakang diatas, maka penulis memberikan solusi dengan merancang alat untuk tugas akhir dengan judul **“Alat Pemberi Makan Ikan Di Akuarium Otomatis Berbasis Mikrokontroler ATmega16”**

B. Identifikasi Masalah

Dari uraian latar belakang di atas, dapat diidentifikasi beberapa permasalahan, antara lain :

1. Saat pemilik pemilik akuarium dalam keadaan sibuk atau dalam berpergian dengan jangka waktu yang lama, tidak ada yang memberi makan ikan-ikannya di akuarium.

2. Dibutuhkan alat yang dapat memberi makan ikan secara otomatis sehingga dapat menjadi solusi bagi pemilik akuarium dengan permasalahan yang telah diuraikan diatas.

C. Batasan Masalah

Karena keterbatasan penulis, maka proyek tugas akhir ini hanya dibatasi pada:

1. Pada alat ini dipasangkan pada akuarium dengan ukuran standar yaitu berbentuk balok persegi panjang dengan ukuran 50x30x30 cm.
2. Pada alat ini hanya meliputi pemberian pakan ikan secara otomatis dan pendeteksi pakan ikan jika dalam kondisi pakan hampir kosong atau habis.

D. Rumusan Masalah

Dari berbagai uraian di atas maka dapat ditarik rumusan masalah, antara lain :

1. Bagaimana merancang dan membangun alat pemberi makan pada ikan di akuarium otomatis berbasis Mikrokontroler ATmega16.
2. Bagaimana unjuk kerja dari alat pemberi makan pada ikan di akuarium otomatis berbasis Mikrokontroler ATmega16.

E. Tujuan

Tujuan dari pembuatan proyek akhir ini antara lain :

1. Merancang dan membangun alat pemberi makan pada ikan di akuarium otomatis berbasis Mikrokontroler ATmega16.
2. Mengetahui unjuk kerja dari alat pemberi makan pada ikan di akuarium otomatis berbasis Mikrokontroler ATmega16.

F. Manfaat

Manfaat yang dapat diambil dari pembuatan proyek akhir ini antara lain :

1. Bagi mahasiswa
 - a. Mahasiswa dapat mengasah kemampuan dalam menciptakan inovasi
 - b. Mahasiswa dapat mengaplikasikan ilmu yang telah diperoleh dalam perkuliahan.
 - c. Mahasiswa dapat membantu menyelesaikan permasalahan di masyarakat.
2. Bagi perguruan tinggi
 - a. Sebagai perwujudan dari tridharma perguruan tinggi yaitu pendidikan, penelitian dan pengabdian masyarakat.
 - b. Untuk menguji mahasiswa dalam menerapkan teori yang telah diperoleh di bangku kuliah ke dalam praktik yang sesungguhnya di masyarakat.
3. Bagi masyarakat
 - a. Memberikan kemudahan pada setiap orang untuk memberikan pakan ikan di akuarium secara otomatis, sehingga orang yang memelihara ikan tersebut walaupun sedang banyak aktivitas, tidak perlu khawatir akan pemberian pakannya.
 - b. Dapat memberikan manfaat bagi pemelihara atau para penjual ikan ketika harus berpergian jauh yang memakan waktu yang cukup lama.

G. Keaslian Gagasan

Proyek ini diilhami dari salah satu hobi penulis dalam memelihara ikan dan akuarium yang dimiliki penulis, sehingga ketika dalam keadaan sibuk atau dalam keadaan berpergian jauh akan terdapat masalah dalam pemberian pakan pada ikan di akuariumnya. Maka dari kejadian tersebut penulis memikirkan untuk membuat alat pemberi makan ikan di akuarium secara otomatis agar mempermudah dalam kegiatan sehari-hari dan juga penulis dapat menuangkan ilmu yang telah dipelajari yaitu yang berkaitan dengan sistem otomatisasi. Dengan alat pemberi makan ikan otomatis ini dapat memberikan manfaat dan menjadi solusi dari permasalahan yang telah dibahas.

Pada proyek akhir ini penulis juga mengacu pada proyek akhir mahasiswa bernama Aditia Putra Kurniawan yang berjudul “*Smart Aquarium*”. Proyek akhir tersebut memiliki fitur yaitu pembersih otomatis dan pengganti air otomatis pada akuarium. Dalam proyek akhir itu menjelaskan bahwa terdapat kekurangan pada alat yang telah dibuat yaitu tidak memiliki fitur pemberi makan secara otomatis yang dijadikan saran untuk pelengkap pada proyek akhir tersebut. Sehingga penulis menjadikan Alat Pemberi Makan Ikan Di Akuarium Otomatis Berbasis Mikrokontroler ATmega16 sebagai pelengkap pada proyek akhir yang berjudul “*Smart Aquarium*”.

BAB II

PENDEKATAN PEMECAHAN MASALAH

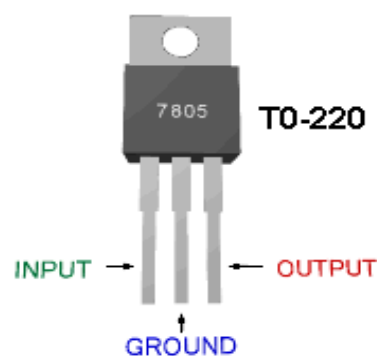
A. Catu Daya

Catu daya merupakan pemberi sumber daya bagi perangkat elektronika. Perangkat elektronika mestinya dicatu oleh *power supply* arus searah (DC) yang stabil agar dapat dengan baik (Zaki.M.H, 2008). Baterai adalah sumber catu daya DC yang paling baik. Namun untuk aplikasi yang membutuhkan catu daya lebih besar, sumber dari baterai tidak cukup. Sumber catu daya yang besar adalah sumber bolak-balik (AC) dari pembangkit tenaga listrik. Untuk itu diperlukan suatu perangkat catu daya yang dapat mengubah arus AC menjadi DC.

Supply daya atau tegangan catu suatu rangkaian elektronik yang berubah-ubah besarnya dapat menyebabkan pengaruh yang sifatnya merusak fungsi kerja rangkaian elektronik yang dicatunya. Oleh sebab itu, jika dari suatu rangkaian elektronik diharapkan suatu kinerja yang prima dan tahan lama, salah satu syaratnya adalah menggunakan catu daya yang stabil dan mampu menekan riak (*ripple*) semaksimal mungkin. Catu daya yang stabil dan dapat diatur sering disebut dengan *regulated power supply*. Catu daya ini menggunakan komponen aktif sehingga harganya cukup mahal. Maka dari itu saat ini banyak digunakan catu daya dalam bentuk IC yaitu IC regulator tegangan. IC regulator tegangan secara garis besar dapat dibagi menjadi dua,

yakni regulator tegangan tetap (3 kaki) dan tegulator tegangan yang dapat diatur (3 kaki atau lebih).

IC regulator tegangan tetap yang sekarang populer adalah keluarga 78xx untuk tegangan positif dan seri 79xx untuk tegangan negatif. Bentuk IC dan susunan kakinya adalah seperti terlihat pada Gambar 1 berikut ini:



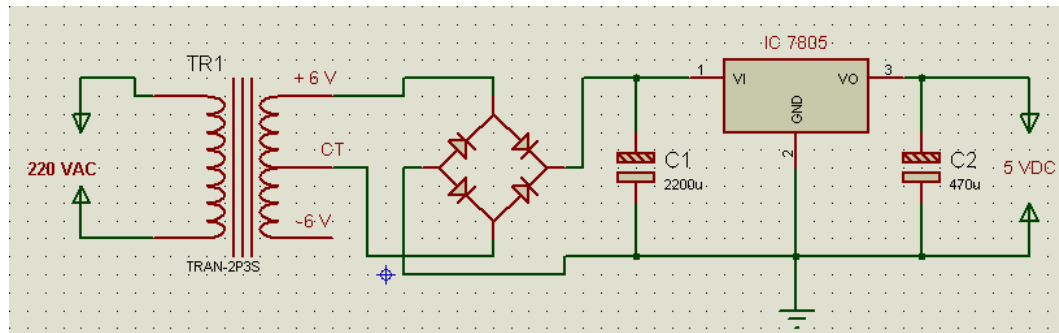
Gambar 1. Konfigurasi Pin pada LM 78xx
(Sumber :Sunomo, 1996)

Tabel 1. Pin Description: 78xx

PinNo	Function	Name
1	Input voltage (5V-18V)	Input
2	Ground (0V)	Ground
3	Regulated output; 5V (4.8V-5.2V)	Output

Besarnya tegangan keluaran IC seri 78xx dan 79xx ini dinyatakan dengan dua angka terakhir pada serinya. Contoh IC 7805 adalah regulator tegangan positif dengan tegangan keluaran 5 volt. Batasan nilai tegangan masukan IC regulator yang terdapat dalam tabel adalah nilai DC. Rangkaian

Catu Daya 5V gelombang penuh dengan menggunakan IC regulator 7805 dapat dilihat pada Gambar 2 berikut:



Gambar 2. Rangkaian Catu Daya 5 Volt
(Sumber: Sutrisno, 2006:94)

Trafo menghasilkan tegangan 6 VAC yang disearahkan dengan diode D1 dan D2, sehingga menghasilkan gelombang penuh, tegangan DC, yang distabilkan dengan bantuan kapasitor C1. Untuk mendapatkan regulasi yang lebih baik dengan tegangan regulasi 5 VDC, maka digunakan IC 7805. Kapasitor C2 digunakan untuk menjaga kestabilan tegangan keluaran dan berfungsi sebagai filter.

B. Trafo Step Down

Transformator atau trafo adalah suatu alat untuk memindahkan enenrgi listrik tanpa hubungan listrik secara langsung dari suatu rangkaian ke rangkaian lain, dengan perubahan tegangan atau arus (Feri andang, 2007:23). *Transformator* terdiri dari 3 komponen pokok yaitu: kumparan pertama (kumparan primer) yang berfungsi sebagai *input*, kumparan kedua (kumparan

skunder) yang berfungsi sebagai output, dan inti besi yang berfungsi untuk memperkuat medan magnet yang dihasilkan.

Transformator bekerja berdasarkan prinsip induksi elektromagnetik. Tegangan masukan bolak-balik yang melewati kumparan primer menimbulkan fluks magnet yang idealnya semua bersambung dengan lilitan sekunder. Fluks bolak – balik ini menginduksikan GGL dalam lilitan sekunder (Sunyoto, 1993). Jika efisiensi sempurna, semua daya pada lilitan primer akan di limpahkan ke lilitan sekunder. Hubungan antara tegangan primer (V_p), jumlah lilitan primer (N_p), tegangan sekunder (V_s), dan jumlah lilitan sekunder (N_s), Perbandingan tegangan pada sisi sekunder dengan tegangan pada sisi primer sama dengan perbandingan jumlah lilitan pada sisi sekunder dan sisi primer seperti persamaan berikut (Sunyoto, 1993).

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

Dengan :

V_P = Tegangan primer (volt)

V_S = Tegangan sekunder (volt)

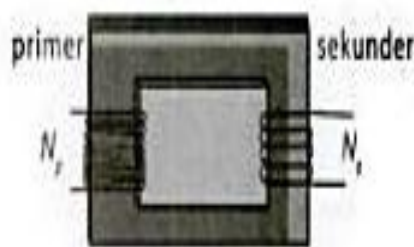
N_P = Jumlah lilitan primer

N_S = Jumlah lilitan sekunder

Berdasarkan perbandingan antara jumlah lilitan primer dan jumlah lilitan sekunder *transformator* ada dua jenis yaitu:

1. *Transformator step up* yaitu *transformator* yang mengubah tegangan bolak-balik rendah menjadi tegangan bolak-balik tinggi, *transformator* ini mempunyai jumlah lilitan kumparan sekunder lebih banyak daripada jumlah lilitan primer ($N_s > N_p$).
2. *Transformator step down* yaitu *transformator* yang mengubah tegangan bolak-balik tinggi menjadi tegangan bolak-balik rendah, *transformator* ini mempunyai jumlah lilitan kumparan primer lebih banyak daripada jumlah lilitan sekunder ($N_p > N_s$).

Rumus perbandingan tersebut menjelaskan bahwa besar tegangan dapat dinaikkan dengan menambah jumlah lilitan pada sisi sekunder dan sebaliknya tegangan dapat diturunkan dengan mengurangi jumlah lilitan pada sisi sekunder. Berikut adalah *transformator stepdown* yang berfungsi menurunkan tegangan tinggi menjadi tegangan rendah dapat dilihat pada Gambar 3.



Gambar 3. Gambar *Transformator Stepdown*
(Sumber : Sunyoto, 1993)

Sedangkan simbol *transformator stepdown* dapat dilihat pada Gambar 4 yaitu jumlah lilitan primer(N_p) lebih banyak dibandingkan jumlah lilitan sekunder(N_s).



Gambar 4. Simbol *Transformator Step Down*

(Sumber: Zaki.M.H, 2008)

C. Driver Solenoid

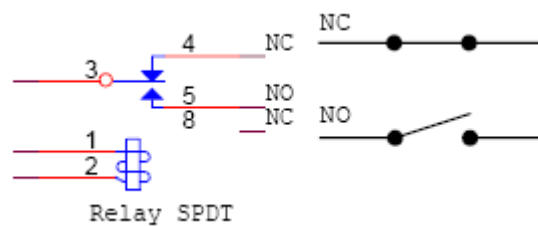
Driver solenoid adalah sebuah rangkaian elektronik yang berfungsi untuk mengatur *solenoid* dari kondisi *on* atau *of* tanpa adanya campur tangan manusia dalam proses pengaturan tersebut. Pada driver solenoid ini terdiri dari beberapa komponen aktif dan tidak aktif. Komponen tersebut adalah :

1. Relay

Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip, relay merupakan tuas saklar dengan lilitan kawat pada batang besi (*solenoid*).

Relay merupakan sebuah piranti elektro mekanik yang dioperasikan berdasarkan variasi masukan, untuk mengontrol piranti-piranti lain yang dihubungkan pada keluaran *relay*. *Relay* berfungsi untuk memutuskan atau mengalirkan arus listrik yang dikontrol dengan

memberikan tegangan suplai pada koilnya. Ada dua jenis *relay* berdasarkan tegangan untuk menggerakkan koilnya, yaitu *relay DC* dan *relay AC*. Pada rangkaian ini menggunakan *relay DC* dengan tegangan 5 volt. simbol dari *relay* ditunjukkan pada Gambar 5 berikut ini:



Gambar 5. Simbol diagram *relay*

(Sumber: Afrie setiawan, 2011)

Kontak-kontak ini dapat digunakan mengontrol arus yang lebih besar dalam rangkaian. Fungsi utama *relay* adalah untuk mengontrol arus yang lebih besar dalam rangkaian dengan arus kecil yang melewati koil *relay*. Pada simbol diatas terdiri atas sebuah kumparan dan dua set kontak, satu diantaranya terbuka *Normally Open (NO)*, dan lainnya tertutup *Normally Close (NC)*. Sewaktu ada tegangan suplai pada koil *relay*, maka kontak *NO* akan terhubung dan kontak *NC* akan terbuka. Sebaliknya saat tidak ada suplai pada koil *relay* maka kontak *NO* kembali terbuka dan kontak *NC* kembali terhubung.

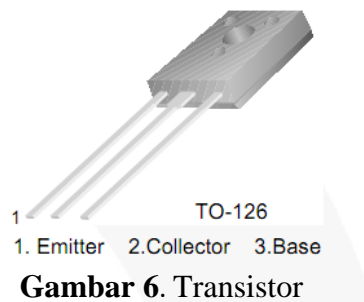
2. Transistor

Transistor merupakan komponen aktif dengan besar arus dan tegangan atau daya keluaran dikendalikan oleh arus masukan. Transistor mempunyai tiga kaki elektroda, yaitu : basis, kolektor dan emitor (*base*,

collector and emitter) (Zaki.M.H, 2008). Transistor dibagi menjadi dua tipe yaitu sambungan *bipolar* atau disebut *Bipolar Junction Transistor (BJT)* dan transistor tipe efek medan atau *Field Effect Transistor (FET)*. Transistor dari tipe sambungan *bipolar* merupakan transistor yang banyak digunakan dalam aplikasi elektronika.

Secara prinsip, transistor sambungan *bipolar* dapat dipahami sebagai sebuah sambungan (*junction*) antara dua buah dioda PN yang saling bertolak belakang. Dua buah dioda tersebut adalah dioda emitor-basis atau disebut dioda emitor dan dioda kolektor-basis atau disebut dioda kolektor. Susunan dari dua buah dioda PN ini menentukan jenis dari transistor. Transistor *bipolar* jenis NPN ini memiliki tiga buah elektroda yang masing-masing disebut dengan emitor atau *emitter (E)*, basis atau *base (B)*, dan kolektor atau *collector (C)*.

Transistor NPN dapat bekerja jika ada bias maju (*forward bias*) yang diberikan padanya. Bias maju merupakan proses pembuatan tegangan pada bahan penyusun transistor sehingga jenis P lebih positif dari pada jenis bahan N. Adanya bias maju ini memungkinkan adanya aliran elektron dari emitor ke kolektor dan arus mengalir dari kolektor ke emitor.

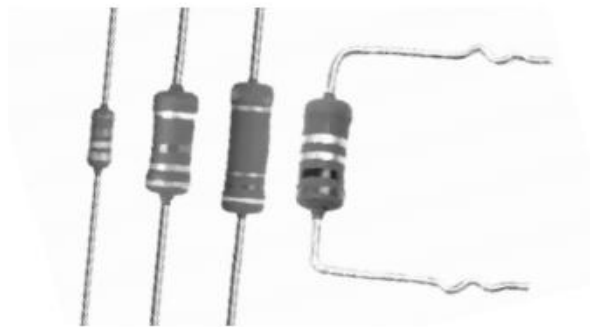


Gambar 6. Transistor

(Sumber: Datasheet BD139)

3. Resistor

Resistor adalah salah satu komponen elektronika yang berfungsi sebagai penahan arus yang mengalir dalam suatu rangkaian dan berupa terminal dua komponen elektronik yang menghasilkan tegangan pada terminal yang sebanding dengan arus listrik yang melewatinya sesuai dengan hukum Ohm ($V = IR$). Sebuah resistor tidak memiliki kutub positif dan negatif, tapi memiliki karakteristik utama yaitu resistensi, toleransi, tegangan kerja maksimum dan power rating. Karakteristik lainnya meliputi koefisien temperatur, kebisingan, dan induktansi. Ohm yang dilambangkan dengan simbol Ω (Omega) merupakan satuan resistansi dari sebuah resistor yang bersifat resistif.



Gambar 7. Resistor

(Sumber: Datasheet Resistor)

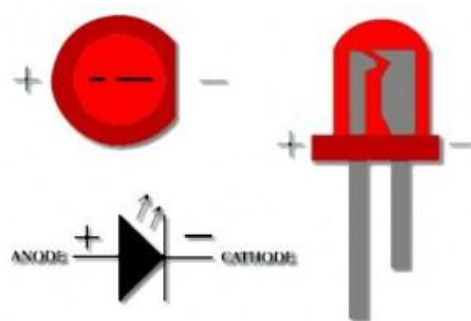
4. LED (*light-emitting diode*)

Diode cahaya atau lebih dikenal dengan nama LED (*light-emitting diode*) adalah dioda yang dibuat dari bahan Ga (*Galium*), As, dan Fosfor yang dapat mengeluarkan emisi cahaya. LED mempunyai struktur yang sama dengan dioda. LED dibuat agar lebih efisien mengeluarkan cahaya. Untuk mendapatkan efisiensi cahaya pada semikonduktor, *doping* yang

dipakai adalah *galium*, *arsenic*, dan *phosporus*. Jenis *doping* yang berbeda akan menghasilkan warna cahaya yang berbeda pula. Contohnya, yaitu :

1. Ga As (*Galium Arsenide*) meradiasikan sinar infrared merah.
2. Ga As P (*Galium Arsenide Phospide*) meradiasikan warna merah dan kuning.
3. Ga P (*Galium Phospide*) meradiasikan warna merah dan kuning

Pada dasarnya semua warna bisa dihasilkan. Dalam memilih warna LED, harus memerhatikan tegangan kerja, arus maksimum, dan disipasi dayanya. (Sumber : Efy Zamidra Zam, 2002). LED memiliki kaki dua buah seperti dengan dioda yaitu kaki anoda dan kaki katoda. Kaki anoda memiliki ciri fisik lebih panjang dari kaki katoda seperti pada gambar dibawah. Pemasangan LED agar dapat menyala adalah dengan memberikan tegangan positif ke kaki anoda dan tegangan negatif ke kaki katoda.



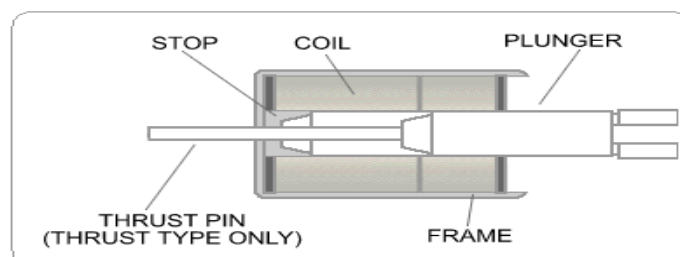
Gambar 8. LED

(Sumber: <http://elektronika-dasar.web.id/komponen/led-light-emitting-dioda/>)

D. Solenoid

Solenoid adalah salah satu jenis kumparan terbuat dari kabel panjang yang dililitkan secara rapat dan dapat diasumsikan bahwa panjangnya jauh lebih besar daripada diameternya. Dalam kasus *solenoid ideal*, panjang kumparan adalah tak hingga dan dibangun dengan kabel yang saling berhimpit dalam lilitannya, dan medan magnet di dalamnya adalah seragam dan paralel terhadap sumbu solenoid.

Jika terdapat batang besi dan ditempatkan sebagian panjangnya di dalam solenoid, batang tersebut akan bergerak masuk ke dalam solenoid saat arus dialirkan. Hal ini dapat dimanfaatkan untuk menggerakkan tuas, membuka pintu, atau mengoperasikan *relay*.



Gambar 9. Bagian-bagian Solenoid

(Sumber: http://www.electronics-tutorials.ws/io/io_6.html)

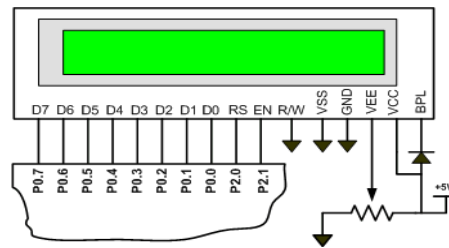


Gambar 10. Solenoid

(Sumber: http://www.electronics-tutorials.ws/io/io_6.html)

E. *Liquid Crystal Display (LCD)*

LCD merupakan salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik yang pengoperasiannya menggunakan sistem *dot* matriks. Tampilan LCD sudah tersedia dalam bentuk modul yaitu tampilan LCD beserta rangkaian pendukungnya. LCD mempunyai pin data, kontrol, catu daya dan pengatur kontras tampilan (Andrianto,2008:69). Konfigurasi Pin LCD yang biasa dipakai dalam menampilkan suatu data dari mikrokontroler berukuran 16x2 karakter. Konfigurasi pin LCD 16x2 karakter adalah seperti pada gambar 11 berikut:



Gambar 11. Konfigurasi Pin LCD Karakter 16x2

(Andrianto,2008:69)

Fungsi pin-pin pada komponen LCD 2x16 ditunjukkan pada tabel 2 berikut ini:

Tabel 2. Konfigurasi Pin LCD

<i>Pin No</i>	<i>Name</i>	<i>Function</i>	<i>Description</i>
1	Vss	<i>Power</i>	GND
2	Vdd	<i>Power</i>	+ 5 V
3	Vee	<i>Contras Adj.</i>	(-2) 0 -5 V
4	RS	<i>Command</i>	<i>Register Select</i>
5	R/W	<i>Command</i>	<i>Read / Write</i>

6	E	<i>Command</i>	<i>Enable (Strobe)</i>
7	D0	I/O	<i>Data</i>
8	D1	I/O	<i>Data</i>
9	D2	I/O	<i>Data</i>
10	D3	I/O	<i>Data</i>
11	D4	I/O	<i>Data</i>
12	D5	I/O	<i>Data</i>
13	D6	I/O	<i>Data</i>
14	D7	I/O	<i>Data</i>

F. Push Button

Push button merupakan saklar yang di operasikan secara manual. *Push button* ini berfungsi untuk memutuskan atau menghubungkan aliran listrik. Ada dua macam *push button*, yaitu *push button Normaly Open (NO)* dan *push button Normaly Close (NC)* (Sumber: Efvy Zamidra Zam, 2002).

Push button NO menghubungkan rangkaian ketika ditekan dan kembali keposisi terbuka ketika dilepas. Sebaliknya *push button NC* membuka rangkaian ketika *push button* ditekan dan kembali pada posisi menutup ketika *push button* dilepas.



Gambar 12. Push Button
(Sumber: Datasheet Push Button)

G. Keypad

Keypad adalah saklar-saklar *push button* yang disusun secara *matriks* yang berfungsi untuk menginput data seperti, *input* pintu otomatis, *input* absensi, *input datalogger*, dan sebagainya. Pada proyek akhir ini *keypad* digunakan sebagai alat untuk memasukkan nilai yang diinginkan pengguna dalam mengatur waktu buka pakan ikan dan waktu untuk *countdown*. *Keypad* dihubungkan ke PORTC mikrokontroler ATmega16.

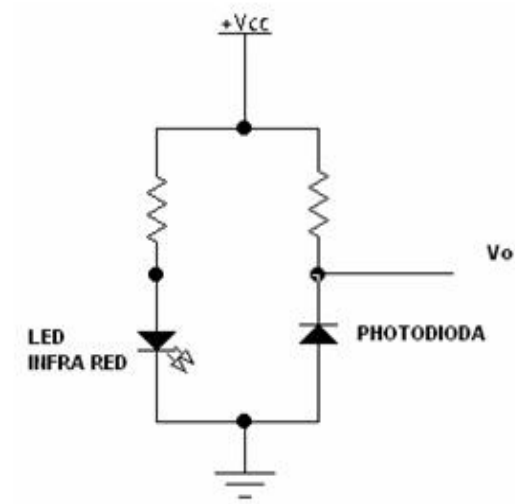


Gambar 13. Gambar *Keypad 3x4*
(Sumber: Koleksi Pribadi)

H. Sensor Objek Benda

Sensor objek benda merupakan sebuah sensor yang terdiri dari beberapa komponen utama yaitu: pemancar (*transmitter*) dan penerima (*receiver*). Sensor Objek benda berfungsi sebagai alat untuk mendeteksi benda yang berada tepat diantara pemancar (*transmitter*) dan penerima (*receiver*), sinar pemancar (*transmitter*) terhalang oleh benda sehingga logika dari penerima (*receiver*) berlogika low.

Pada sensor benda ini komponen yang digunakan sebagai pemancar (*transmitter*) adalah LED, dan penerima (*reciever*) adalah photo dioda.



Gambar 14. Rangkaian Sensor Benda
(Sumber : Heri Andrianto, 2008 : 3)

1. Photo Dioda

Komponen ini akan mengubah energi cahaya, dalam hal ini energi cahaya infra merah menjadi sinyal listrik (dalam hal ini arus listrik). Merupakan sambungan dioda PN yang memiliki kepekaan terhadap radiasi gelombang Elektromagnetik (EM) ketika jatuh pada sambungan. Dikarenakan sambungan PN sangatlah kecil, dibutuhkan lensa untuk memfokuskan radiasi yang datang agar mendapatkan respon yang baik. Keunggulan *device* ini adalah nilai waktu responnya sangatlah cepat. Kebanyakan memiliki waktu respon mendekati 1 Mikrodetik, bahkan ada yang mendekati 1 nano detik. Semakin tinggi intensitas cahaya, maka arus

bocor pada sambungan PN semakin besar sehingga arus yang lewat sambungan semakin kecil.

2. LED

Pada rangkaian sensor benda diatas terdiri dari 2 bagian, yaitu bagian pemancar cahaya dan penerima cahaya. Rangkaian pemancar terdiri dari resistor sebagai pembatas arus serta LED sebagai piranti yang memancarkan cahaya. Sedangkan rangkaian penerima terdiri dari resistor sebagai *pull-up* tegangan dan photodioda sebagai piranti yang akan menerima cahaya LED. LED merupakan komponen yang dapat mengeluarkan emisi cahaya. Struktur pada LED hampir sama dengan struktur pada dioda, tetapi ada penemuan bahwa elektron yang menerjang sambungan PN juga melepaskan energi berupa energi panas dan energi cahaya. LED dibuat agar lebih efisien jika mengeluarkan cahaya.

I. Mikrokontroler ATmega16

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu serpih (*chip*). Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan *Read-Only Memory* (ROM), *Read-Write Memory* (RAM), beberapa port masukan maupun keluaran, dan beberapa *peripheral* seperti pencacah/pewaktu, *Analog to Digital converter* (ADC), *Digital to Analog Converter* (DAC) dan serial komunikasi. (Heri Andrianto, 2008:69)

Salah satu mikrokontroler yang banyak digunakan saat ini yaitu mikrokontroler AVR. AVR adalah mikrokontroler *Reduce Instruction Set Computer* (RISC) 8 bit berdasarkan arsitektur Harvard. Secara umum mikrokontroler AVR dapat dikelompokkan menjadi 3 (tiga) kelompok, yaitu keluarga AT90Sxx, ATmega dan ATtiny. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fiturnya: (Heri Andrianto, 2008:69)

1. Arsitektur ATmega16

Mikrokontroler ini menggunakan arsitektur Harvard yang memisahkan memori program dari memori data, baik bus alamat maupun bus data, sehingga pengaksesan program dan data dapat dilakukan secara bersamaan (*concurrent*). Secara garis besar mikrokontroler ATmega16 terdiri dari :

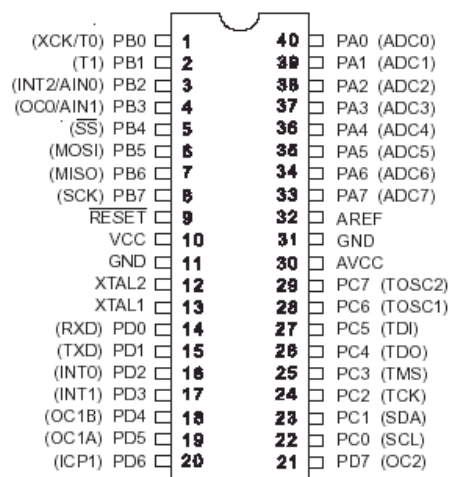
- a. Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi, dengan daya rendah.
- b. Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16Mhz.
- c. Memiliki kapasitas *Flash* memori 16Kbyte, EEPROM 512 Byte, dan SRAM 1Kbyte.
- d. Saluran I/O 32 buah, yaitu *Port A*, *Port B*, *Port C* dan *Port D*.
- e. CPU yang terdiri dari 32 buah *register*.
- f. Unit interupsi internal dan eksternal
- g. *Port* USART sebagai komunikasi serial.

h. Fitur *Peripheral*

- ♦Tiga buah *timer/counter* dengan kemampuan perbandingan.
- ♦*Real time counter* dengan osilator tersendiri
- ♦Empat kanal PWM dan antarmuka komparator analog
- ♦8 kanal, 10 bit ADC
- ♦*Byte-oriented Two-wire Serial Interface*
- ♦*Programmable Serial USART*
- ♦Antarmuka SPI
- ♦*Watchdog timer* dengan osilator internal
- ♦*On-chip analog comparator*

2. Konfigurasi Pin ATmega16

Konfigurasi pin mikrokontroler ATmega16 dengan kemasan 40 pin dapat dilihat pada gambar di bawah ini. Dari gambar tersebut dapat terlihat ATmega16 memiliki 8 pin untuk masing-masing *Port A*, *Port B*, *Port C*, dan *Port D*, Seperti terlihat seperti gambar 8 dibawah ini.



Gambar 15. Konfigurasi Pin ATmega16
(Andrianto,2008:69)

Konfigurasi pin ATmega16 dengan kemasan 40 pin *Dual In-line Package* (DIP) dapat dilihat pada Gambar 15. Dari gambar di atas dapat dijelaskan fungsi dari masing-masing pin ATmega16 sebagai berikut:

- a. Vcc merupakan pin yang berfungsi sebagai masukan catu daya.
- b. GND merupakan pin *ground*
- c. Port A (PA0...7) merupakan pin *input/output* dua arah dan pin masukan ADC.
- d. Port B (PB0...7) merupakan pin *input/output* dua arah dan pin dengan fungsi khusus seperti SPI, MISO, MOSI, SS, AIN1/OC0, AIN0/INT2, T1, T0 T1/XCK

Tabel 3. Fungsi Khusus masing-masing *pin* pada *PORT B*

Pin	Fungsi Khusus
PB7	<i>SCK (SPI Bus Serial Clock)</i>
PB6	<i>MISO (SPI Bus Master Input/ Slave Output)</i>
PB5	<i>MOSI (SPI Bus Master Output/ Slave Input)</i>
PB4	<i>\overline{SS} (SPI Slave Select Input)</i>
PB3	<i>AIN1 (Analog Comparator Negative Input)</i> <i>AC0 (Timer/ Counter0 Output Compare Match Output)</i>
PB2	<i>AIN1 (Analog Comparator Positive Input)</i> <i>INT2 (External Interrupt 2 Input)</i>
PB1	<i>T1 (Timer/ Counter 1 External Counter Input)</i>
PB0	<i>T0 T1 (Timer/ Counter0 External Counter Input)</i> <i>XCK (USART External Clock Input Output)</i>

- e. Port C (PC0...7) merupakan pin *input/output* dua arah dan pin dengan fungsi khusus, seperti TOSC2, TOSC1, TDI,TD0, TMS, TCK, SDA, SCL

Tabel 4. Fungsi Khusus masing-masing *pin* pada *PORT C*

Pin	Fungsi Khusus
<i>PC7</i>	<i>TOSC2 (Timer Oscillator Pin2)</i>
<i>PC6</i>	<i>TOSC1 (Timer Oscillator Pin)</i>
<i>PC5</i>	<i>TDI (JTAG Test Data In)</i>
<i>PC4</i>	<i>TDO (JTAG Test Data Out)</i>
<i>PC3</i>	<i>TMS (JTAG Test Mode Select)</i>
<i>PC2</i>	<i>TCK (JTAG Test Clock)</i>
<i>PC1</i>	<i>SDA (Two-wire Serial Bus Data Input/ Output Line)</i>
<i>PC0</i>	<i>SCL (Two-wire Serial Bus Clock Line)</i>

- f. Port D (PD0...7) merupakan pin *input/output* dua arah dan pin dengan fungsi khusus, seperti RXD, TXD, INT0, INT1, OC1B, OC1A, ICP1.

Tabel 5. Fungsi Khusus masing-masing *pin* pada *PORT D*

Pin	Fungsi Khusus
<i>PD7</i>	<i>OC2 (Timer/ Counter2 Output Compare Match Output)</i>
<i>PD6</i>	<i>ICP (Timer/ Counter1 Input Capture Pin)</i>
<i>PD5</i>	<i>OC1A (Timer/ Counter1 Output Compare A Match Output)</i>
<i>PD4</i>	<i>OC1B (Timer/ Counter1 Output Compare B Match Output)</i>
<i>PD3</i>	<i>INT1 (External Interrupt 1 Input)</i>
<i>PD2</i>	<i>INT0 (External Interrupt 0 Input)</i>




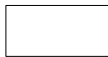


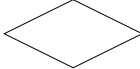


<i>PD1</i>	<i>TXD (USART Output Pin)</i>
<i>PD0</i>	<i>RXD (USART Input Pin)</i>

- g. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
- h. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal
- i. AVCC merupakan pin masukan tegangan untuk ADC
- j. AREF merupakan pin masukan tegangan referensi ADC.

J. Diagram Alir (*Flowchart*)

Dalam merancang sebuah program, pembuat menganggap sebuah program rancangannya sudah selesai jika program tersebut telah berjalan sesuai dengan yang diharapkan. Program yang dirancang perlu ditelusuri lagi untuk keperluan pengembangan lebih lanjut dari cara kerja program rancangan tersebut. Teknik rancang sebuah program dengan struktur yang baik biasanya diawali dengan pembuatan diagram alir (*flowchart*). Diagram alir (*flowchart*) adalah penyajian yang sistematis tentang proses logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Sistem pada diagram alir adalah urutan proses dalam sistem yang menunjukkan alat media *input*, output serta jenis media penyimpanan dalam proses pengolahan data. Diagram alir digunakan untuk menggambarkan terlebih dahulu mengenai apa yang harus dikerjakan sebelum mulai merancang program. Berikut simbol-simbol diagram alir ditunjukkan pada tabel 6 berikut ini : (*sumber: Dian Artanto, 2007*).

Tabel 6. Simbol-simbol dalam diagram alir

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Mulai/selesai
	GARIS ALIR(FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi
	PROSES	proses pengolahan data
	INPUT/OUTPUT DATA	Proses <i>input/output</i> data
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan penyeleksian data langkah selanjutnya
	ON PAGE CONNECTION	Penghubung bagian <i>flowchart</i> yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian <i>flowchart</i> yang berada pada halaman berbeda

BAB III

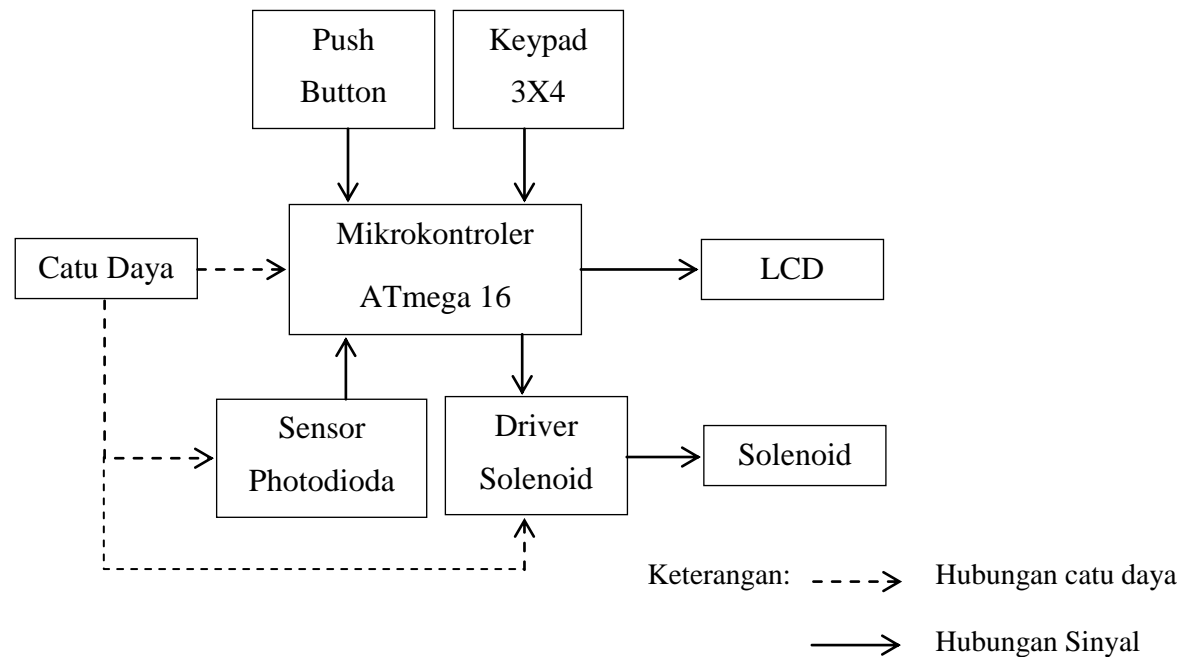
KONSEP PERANCANGAN ALAT

Perancangan proyek akhir ini mempunyai beberapa langkah yaitu analisis kebutuhan, perancangan perangkat keras, perancangan perangkat lunak dan perencanaan pengambilan data.

A. Analisis Kebutuhan

“Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16” secara garis besar memiliki fitur untuk membuka dan menutup tempat pakan ikan secara otomatis dan juga dapat ditentukan pengaturan waktunya oleh pengguna. Dalam fitur ini menggunakan IC mikrokontroler ATmega16 sebagai pusat pengendalian yang telah diisi program sebelumnya.

Secara garis besar alat pemberi makan ini memiliki tiga jenis pengaturan antara lain dengan cara manual, semi-otomatis dan otomatis. Langkah pertama pengguna memilih dari tiga jenis pengaturan dengan menekan *push button* yang terdapat pada alat. Pada pengaturan manual pengguna menggunakan *push button* sebagai kontrolnya. Sedangkan pada pengaturan semi otomatis dan *full* otomatis pengguna menggunakan *keypad* untuk mengatur waktu yang akan ditentukan oleh pengguna. Untuk memudahkan perancangan, maka dibuat diagram blok yang dapat dilihat pada gambar 16.



Gambar 16. Diagram Blok Rangkaian

Keterangan dari diagram blok pada gambar 16 adalah sebagai berikut :

1. Catu Daya sebagai input rangkaian atau penyuplai tegangan, yaitu 5 Volt dan 12 Volt.
2. IC Mikrokontroler yang digunakan adalah IC Mikrokontroler ATmega16 yang bekerja pada tegangan 5 volt dan berfungsi sebagai kontrol rangkaian.
3. *Push Button* sebagai *input* untuk mengatur jenis pengaturan yang akan digunakan oleh pengguna yang terdiri dari tombol manual, tombol semi-auto dan tombol full-auto.
4. *Keypad* sebagai input untuk mengatur waktu buka solenoid dan waktu countdown yang akan ditentukan oleh pengguna.

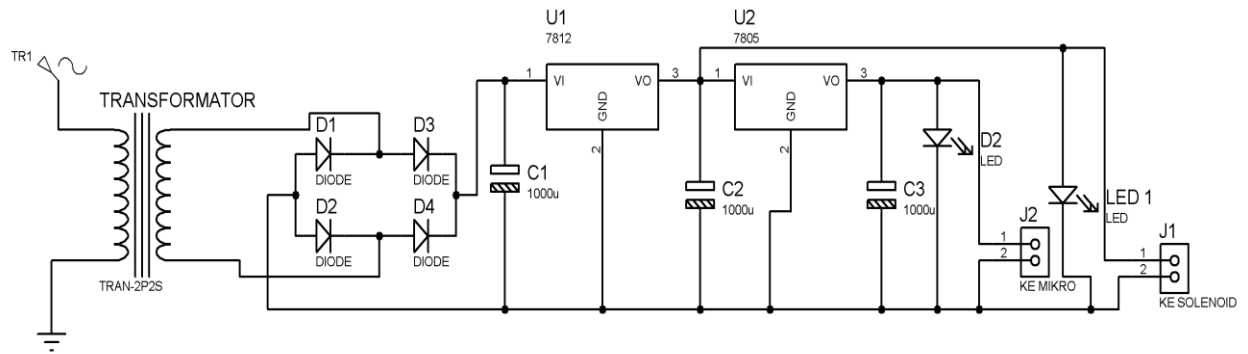
5. Sensor Photodiode sebagai input untuk mendeteksi pakan ikan dalam wadah dalam keadaan isi atau kosong.
6. Driver Solenoid bekerja pada tegangan 12 volt berfungsi sebagai saklar untuk mengaktifkan solenoid.
7. Solenoid berfungsi untuk membuka dan menutup tempat pakan ikan pada alat.
8. LCD berfungsi sebagai tampilan dari fitur alat yaitu pemilihan menu, pengaturan waktu buka solenoid dan pengaturan waktu *countdown*.

B. Perancangan Perangkat Keras

“Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16” terdiri dari perancangan perangkat keras yaitu :

1. Perancangan Rangkaian Catu Daya

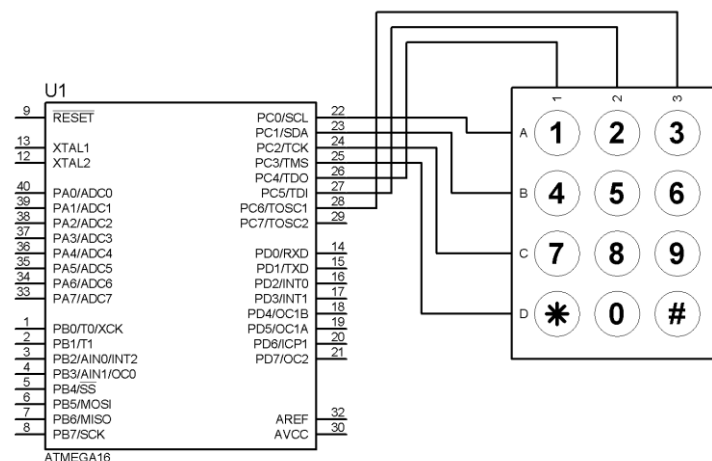
Rangkaian catu daya terdiri dari komponen transformator *stepdown*. Transformator ini mendapatkan suplai dari tegangan PLN, kemudian tegangan tersebut diturunkan dari 220 volt pada sisi primer menjadi 15 volt pada sisi sekunder. Tegangan keluaran dari trafo kemudian disearahkan dengan rangkaian *diode bridge* sehingga menghasilkan tegangan DC yang distabilkan dengan bantuan kapasitor. Untuk mendapatkan regulasi yang lebih baik dengan tegangan 5 VDC dan 12 VDC, maka digunakan IC LM 7805 untuk sumber +5 volt dan IC LM 7812 untuk sumber +12 volt yang selanjutnya juga distabilkan dengan kapasitor yang bertujuan untuk mengurangi riak dan memperhalus tegangan seperti pada gambar 17.



Gambar 17. Rangkaian Catu Daya

2. Perancangan Rangkaian Keypad 3X4

Pemilihan *keypad* 3x4 berdasarkan banyaknya jumlah tombol yang dibutuhkan sebagai input mikrokontroler ATmega16. Pada dasarnya *keypad* hanya tersusun dari beberapa *push button* yang dikonfigurasi antara kolom dan barisnya. Sehingga sering disebut juga *keypad* matriks nxm (n=kolom m=baris). Kolom dan baris ini nantinya yang digunakan untuk pendeteksian penekanan tombol, berikut adalah konfigurasi untuk *keypad* 3x4 dihubungkan dengan PORTB ditunjukkan pada gambar berikut.



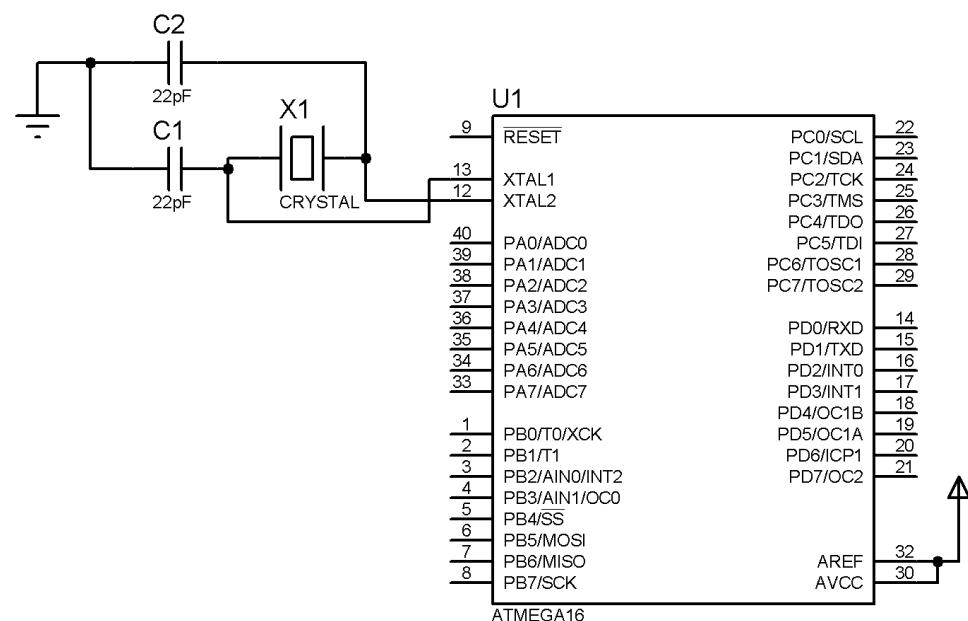
Gambar 18. Rangkaian Keypad 3X4

Fungsi *keypad* dari alat ini yaitu sebagai input untuk memasukan data yang ditentukan oleh pengguna yaitu waktu untuk lamanya buka solenoid dan waktu untuk countdown. Pengguna dapat mengatur dalam

waktu pemberian pakan ikan dengan waktu nyata yang terdiri dari pilihan jam, menit dan detik.

3. Rangkaian sistem Minimum Mikrokontroler ATmega16

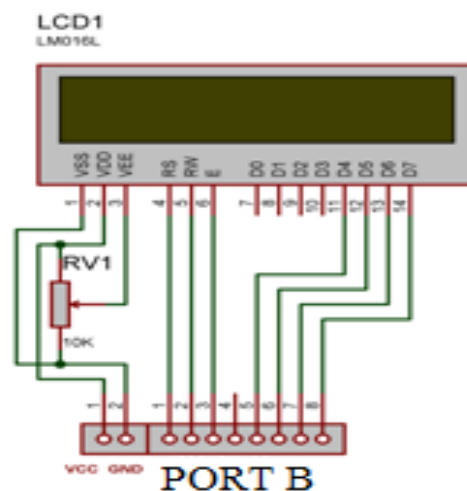
Skema sistem dasar rangkaian sistem minimum mikrokontroler ATmega16 mengacu pada buku “Pemrograman Mikrokontroler AVR ATmega16 Menggunakan Bahasa C” karangan Heri Andrianto. Rangkaian Mikrokontroler ATmega16 ini tersusun dari rangkaian sistem minimum yaitu IC ATmega16, oscilator eksternal dan reset. Oscilator eksternal berfungsi untuk menentukan kecepatan eksekusi program. Rangkaian *oscilator* eksternal terdiri dari komponen 2 buah kapasitor 22pF dan *crystal* dengan nilai 11.0592MHz, tombol reset berfungsi untuk mereset mikrokontroler. PORTA difungsikan sebagai input push button, PORTB difungsikan sebagai output yang di-*interface*-kan dengan LCD dan PORTC difungsikan sebagai input yang di-*interface*-kan dengan *keypad*.



Gambar 19. Skema Rangkaian Sistem Minimum Mikrokontroler ATmega16

4. Rangkaian LCD

LCD pada alat ini difungsikan sebagai penampil dari *keypad* dan juga *push button* dalam pengaturan menu dan waktu. Dalam mengatur menu yaitu dengan menggunakan *push button* sebagai inputannya. Menu yang akan ditampilkannya yaitu pemilihan antara menu “Semi-Auto” dan menu “Auto”. Sedangkan tombol *keypad* untuk melakukan pengaturan waktu dengan memasukkan data(jam, menit dan detik) pada LCD. LCD akan memudahkan dalam pengoprasian tombol karena penekanan tombol bisa langsung ditampilkan dalam LCD. Berikut skema rangkaian LCD monitor 16x2 dihubungkan dengan PORTB ditunjukkan pada Gambar 20 berikut ini :

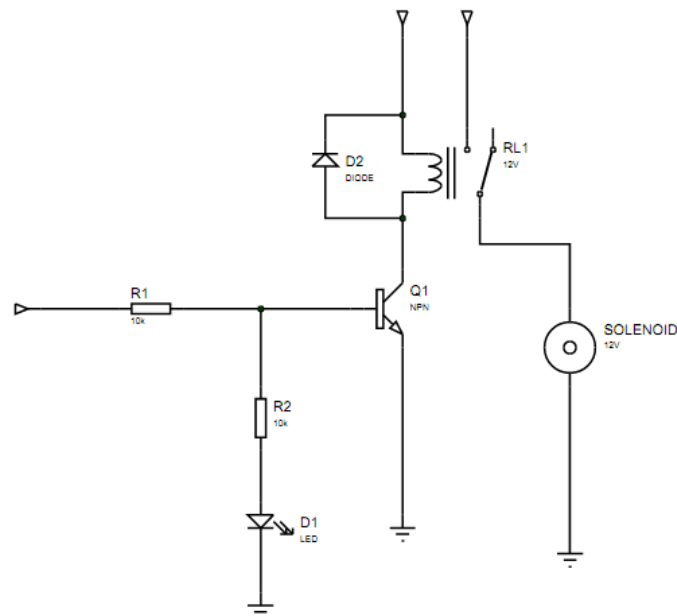


Gambar 20. Skema Rangkaian LCD Monitor 16x2

5. Rangkaian Driver Solenoid

Rangkaian ini berfungsi mengatur aktif tidaknya beban. Beban yang digunakan adalah solenoid. Rangkaian ini mempunyai input 12 Volt. Rangkaian *driver* pada sistem ini menggunakan prinsip kerja transistor

sebagai saklar yang nantinya sebagai masukan *relay*, dimana *driver* ini bekerja apabila ada masukan dari rangkaian pengontrol (Mikrokontroler ATmega16). Jika output dari Mikrokontroler ATmega16 berlogika “1” maka transistor akan berada pada *mode* jenuh (saturasi), sehingga *relay* akan bekerja mengaktifkan aktuator. Rangkaian yang kedua output dari mikrokontroler berlogika “0”, maka transistor akan berada pada *mode* operasi *cut off* (mati) sehingga *relay* tidak aktif.



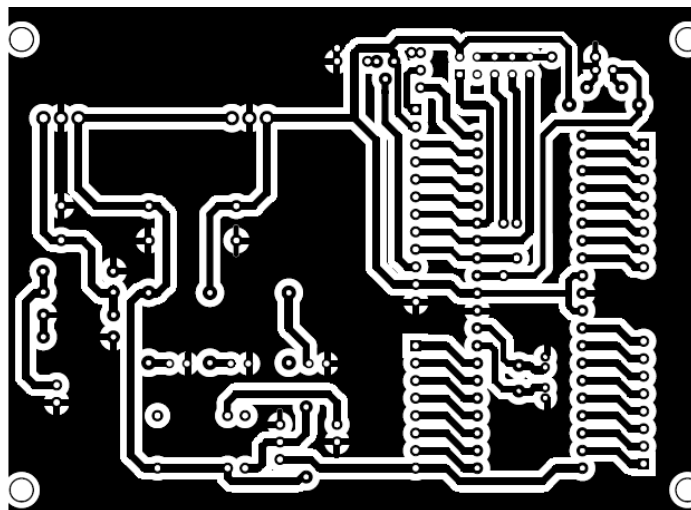
Gambar 21. Rangkaian Driver Solenoid

6. Pembuatan *Printed Circuit Board (PCB)*

Langkah awal pembuatan PCB adalah menggambar *layout* rangkaian dengan perangkat lunak *PCB Wizard 3.50*. Setelah *layout* selesai dibuat maka langkah selanjutnya yaitu menyablonkan *layout* ke PCB polos. Proses penyablonan dilakukan dengan cara :

- a) Mencetak *layout* pada kertas.
- b) Memfotocopy *layout* rangkaian PCB pada kertas glossi.
- c) Desain *layout* yang sudah dicetak pada kertas glossi disablonkan ke PCB dengan cara disetrika selama kurang lebih 10 menit.
- d) Setelah gambar *layout* menempel pada PCB maka kertas yang menempel pada PCB dilepas dengan air sampai bersih.

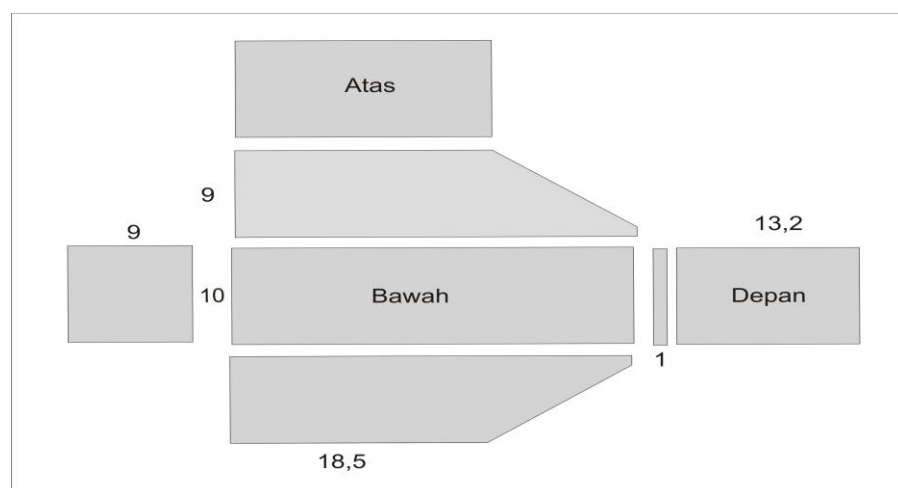
Langkah selanjutnya yaitu melarutkan PCB dengan cairan *feri chloride* dengan cara melarutkan cairan ferichloride dengan air, lalu masukkan papan PCB yang telah tertempel gambar *layout* jalur PCB dan goyang-goyangkan sampai jalur rangkaian terbentuk, kemudian PCB diangkat dari cairan *feri chloride* dan dibersihkan dengan air. Setelah bersih PCB dibor sesuai dengan titik – titik yang telah ditentukan.



Gambar 22. Gambar *Layout* PCB

7. Pembuatan Box

Perencanaan ukuran box rangkaian sebagai berikut panjang 29 cm, lebar 10 cm dan tinggi 9 cm. Pembuatan box ini keseluruhan terbuat dari akrilik. *Box* ini digunakan untuk tempat meletakkan rangkaian catu daya, rangkain mikrokontroller dan rangkaian *relay*. Pada box ini juga terdapat beberapa bagian, yaitu pada bagian depan untuk menempatkan *keypad*, push button dan LCD. Pada bagian dalam terdapat rangkaian-rangkaian elektronik seperti yang disebutkan diatas. Pada bagian atas terdapat kotak yang terbuat dari akrilik juga yang berfungsi untuk tempat pakan ikan dan pada bagian belakang terdapat saklar dan kabel penghubung untuk sumber 220 volt. Proses pembuatan *box* terdiri dari pembuatan desain akrilik, pemotongan akrilik, pengeboran, penggabungan dan pengeleman. Dari proses perencanaan ukuran sebelumnya didapatkan hasil bentuk *box* seperti pada Gambar 23.



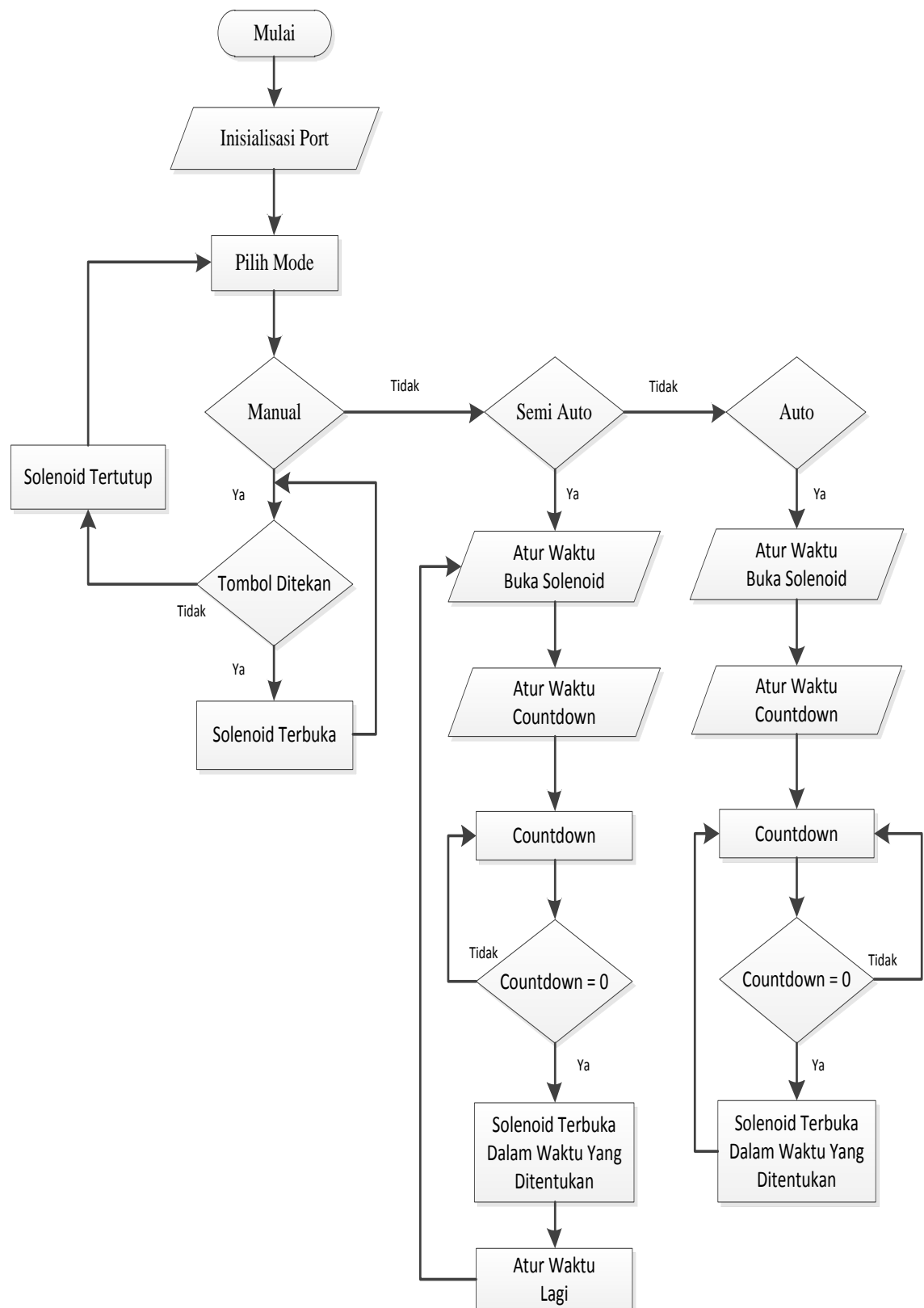
Gambar 23. Sketsa Rancang Bangun Box Alat



Gambar 24. Box Alat tampak keseluruhan

C. Perancangan Perangkat Lunak

Perangkat lunak yang dirancang dengan menggunakan aplikasi Codevision AVR. Untuk memberikan gambaran umum jalannya program dan memudahkan pembuatan perangkat lunak, maka dibuat diagram alir yang menunjukkan jalannya program. Perangkat lunak ini berfungsi untuk mengatur kinerja keseluruhan dari sistem yang terdiri dari beberapa perangkat keras sehingga sistem ini dapat bekerja dengan baik.



Gambar 25. Diagram Alir Program Utama

D. Perencanaan Pengujian Pengambilan Data

Tujuan pengambilan data adalah untuk mengetahui kebenaran rangkaian dan mengetahui kondisi komponen, alat, serta hasil dari pengujian alat itu sendiri.

1. Langkah-langkah Pengambilan Data

- a. Siapkan alat “Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16” yang akan diuji.
- b. Hidupkan alat “Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16”.
- c. Lakukan pengukuran tegangan keluaran pada rangkaian catu daya.
- d. Cek tampilan LCD.
- e. Cek tombol *push button* dengan cara menekannya dan cek apakah tampilan pada LCD sudah sesuai.
- f. Cek tombol *keypad* dengan cara menekannya dan cek apakah tampilan pada LCD sudah sesuai.
- g. Cek tegangan AC pada setiap stop kontak, cocokkan dengan led indikator yang menyala.
- h. Catat hasilnya pada tabel pengujian.

2. Alat dan Bahan yang Digunakan

- a. Multimeter Digital
- b. Timbangan Digital
- c. Stopwatch

3. Perancangan Tabel Pengujian

Pengujian terhadap “Alat Pemberi Makan Ikan Di Akuarium Otomatis Berbasis Mikrokontroler ATmega16” dilakukan pada beberapa bagian. Perancangan tabel pengujian dari masing-masing bagian ditunjukkan pada Tabel berikut :

- a. Pengujian Rangkaian Catu Daya
- b. Pengujian Tampilan LCD
- c. Pengujian Fungsional Alat
- d. Pengujian Ketepatan Waktu Alat
- e. Pengujian Berat Pakan yang Dikeluarkan

Tabel 7. Pengujian Rangkaian Catu Daya

Percobaan Ke-	Pengukuran Tegangan Output Catu Daya	Data		Persentase Kesalahan
		Rekomendasi	Rangkaian	
1	LM 7805			%
	LM 7812			%
2	LM 7805			%
	LM 7812			%

Tabel 8. Pengujian Tampilan LCD

Tampilan LCD		
NO	Penekanan Tombol <i>Keypad</i>	Tampilan Pada LCD
1.	Angka 1	
2.	Angka 2	
3.	Angka 3	
4.	Angka 4	
5.	Angka 5	
6.	Angka 6	
7.	Angka 7	
8.	Angka 8	
9.	Angka 9	
10.	Angka 0	
11.	Tombol *	
12.	Tombol #	

Tabel 9. Pengujian Fungsional Alat

No.	Daftar Uji	Keterangan	
		Bisa	Tidak Bisa
1.	Tampilan LCD		
2.	<i>Keypad</i>		
3.	Tombol Menu		
4.	Tombol Manual		
5.	Tombol Semi-Auto		
6.	Tombol Otomatis		
7.	Solenoid		
8.	Sensor Photodioda		
9.	LED Indikator		

Tabel 10. Pengujian Ketepatan Waktu Alat

No	Percobaan	Waktu alat	Waktu Sesungguhnya	Selisih Waktu
1	1			
	2			
	3			
	4			
	5			
2	1			
	2			
	3			
	4			
	5			

Tabel 11. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu
Buka Solenoid 500 ms

Percobaan	Berat Pakan (gram)
1	
2	
3	
4	
5	

Tabel 12. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu
Buka Solenoid 1000 ms

Percobaan	Berat Pakan (gram)
1	
2	
3	
4	
5	

Tabel 13. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu
Buka Solenoid 1500 ms

Percobaan	Berat Pakan (gram)
1	
2	
3	
4	
5	

BAB IV

HASIL PENGUJIAN DAN PEMBAHASAN

A. Hasil Pengujian Alat

Pengujian alat ini bertujuan untuk mengetahui kebenaran rangkaian dan kesesuaian kerja Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16 dengan fitur yang diinginkan. Pengujian ini terdiri dari pengujian rangkaian catu daya, pengujian tampilan LCD, pengujian fungsional alat, pengujian ketepatan waktu alat dan pengujian berat pakan yang dikeluarkan.

Pengujian catu daya dilakukan untuk mengetahui tegangan yang dihasilkan oleh rangkaian catu daya sudah sesuai dengan yang diharapkan atau tidak menggunakan multimeter digital. Pengujian pada tampilan LCD dilakukan untuk mengetahui sesuai atau tidaknya penekanan *keypad* pada alat dengan tampilan pada LCD. Pengujian fungsional alat yaitu untuk menguji bagian-bagian dari alat pemberi makan ikan ini dapat bekerja atau tidak. Pengujian ketepatan waktu alat dilakukan untuk mengetahui kepresisian waktu pada program dengan waktu yang sebenarnya. Pengujian pada berat pakan dilakukan untuk mengetahui kepresisian berat pakan yang dikeluarkan secara terus menerus, apakah berat pakan yang dikeluarkan itu berbeda atau tidak.

Setelah dilakukan pengujian kerja Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16, didapat data hasil pengujian yaitu pada Tabel 14, Tabel 15, Tabel 16, Tabel 17, Tabel 18, Tabel 19 dan Tabel 20.

Tabel 14. Pengujian Rangkaian Catu Daya

Percobaan Ke-	Komponen	Tegangan Keluaran		Persentase Kesalahan
		Rekomendasi	Pengukuran	
1	LM 7805	5 V	4,99 V	0,2%
	LM 7812	12 V	11,98 V	0,17%
2	LM 7805	5 V	4,99 V	0,2%
	LM 7812	12 V	11,97 V	0,25%

Tabel 15. Pengujian Tampilan LCD

Tampilan LCD		
NO	Penekanan Tombol <i>Keypad</i>	Tampilan pada LCD
1.	Angka 1	1
2.	Angka 2	2
3.	Angka 3	3
4.	Angka 4	4
5.	Angka 5	5
6.	Angka 6	6
7.	Angka 7	7
8.	Angka 8	8
9.	Angka 9	9
10.	Angka 0	0
11.	Tombol *	OK
12.	Tombol #	RESET

Tabel 16. Pengujian Fungsional Alat

No.	Daftar Uji	Keterangan	
		Bisa	Tidak bisa
1.	Tampilan LCD	V	
2.	<i>Keypad</i>	V	
3.	Tombol Menu	V	
4.	Tombol Manual	V	
5.	Tombol Semi-Auto	V	
6.	Tombol Otomatis	V	
7.	Solenoid	V	
8.	Sensor Photodiode	V	
9.	LED Indikator	V	

Tabel 17. Pengujian Ketepatan Waktu Alat

No	Percobaan	Waktu alat	Waktu Sesungguhnya	Selisih Waktu
1	1	00:01:00	00:01:00	00:00:00
	2	00:10:00	00:10:00	00:00:00
	3	00:30:00	00:30:00	00:00:00
	4	01:00:00	01:00:01	00:00:01
	5	02:00:00	02:00:02	00:00:02
2	1	00:01:00	00:01:00	00:00:00
	2	00:10:00	00:10:00	00:00:00
	3	00:30:00	00:30:00	00:00:00
	4	01:00:00	01:00:01	00:00:01
	5	02:00:00	02:00:03	00:00:03

Tabel 18. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 500 ms

Percobaan	Berat Pakan (gram)
1	1,46
2	1,56
3	1,66
4	1,65
5	1,67

Tabel 19. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 1000 ms

Percobaan	Berat Pakan (gram)
1	4,84
2	4,74
3	4,83
4	4,81
5	4,86

Tabel 20. Pengujian Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 1500 ms

Percobaan	Berat Pakan (gram)
1	8,28
2	8,42
3	8,37
4	8,36
5	8,32

B. Pembahasan Hasil Pengujian

1. Catu Daya (*Power Supply*)

Pengamatan dilakukan dengan mengukur tegangan catu daya (*power supply*) menggunakan multimeter digital. Rangkaian catu daya terdiri dari komponen transformator *stepdown*. Transformator ini mendapatkan suplai dari tegangan PLN, kemudian tegangan tersebut diturunkan dari 220V AC pada sisi primer menjadi 15V AC pada sisi sekunder. Tegangan keluaran dari trafo kemudian disearahkan dengan rangkaian *diode bridge* sehingga menghasilkan tegangan DC yang distabilkan dengan bantuan kapasitor. Untuk mendapatkan regulasi yang lebih baik dengan tegangan 5V DC dan 12V DC, maka digunakan IC LM 7805 untuk sumber +5 volt dan IC LM

7812 untuk sumber +12 volt yang selanjutnya juga distabilkan dengan kapasitor yang bertujuan untuk mengurangi riak dan memperhalus tegangan.

Berdasarkan hasil pengujian *power supply*, besarnya tegangan keluaran IC LM7805 adalah 4,99 volt. Idealnya besar tegangan keluaran IC LM7805 adalah 5 volt. Penyimpangan tegangan keluaran sebesar:

$$error (\%) = \frac{|5 - 4,99|}{5} \times 100\% = 0,2\%$$

Berdasarkan hasil pengujian *power supply*, besarnya tegangan keluaran IC LM7812 adalah 11,98 volt. Idealnya besar tegangan keluaran IC LM7812 adalah 12 volt. Penyimpangan tegangan keluaran sebesar:

$$error (\%) = \frac{|12 - 11,98|}{12} \times 100\% = 0,17\%$$

Penyimpangan yang terjadi cukup kecil yaitu sebesar 0,2% dan 0,17%. Penyimpangan ini masih dalam batas toleransi.

2. Pengamatan Tampilan LCD

Pengamatan tampilan LCD dilakukan untuk mengetahui sesuai atau tidaknya penekanan *keypad* pada alat dengan tampilan pada LCD. Tombol *keypad* yang ditekan harus sama dengan tampilan yang muncul pada LCD. Berdasarkan data pengujian pada bagian tampilan LCD yang telah dilakukan dapat disimpulkan bahwa alat ini sudah bekerja dengan baik dan sesuai dengan yang diharapkan.

3. Pengamatan Fungsional Alat

Pada tabel uji fungsional alat terdapat beberapa bagian yang diuji. Dari hasil tersebut didapatkan bahwa semua bagian pada alat dapat bekerja dengan baik. Bagiam-bagian tersebut tidak lepas dari program yang di-downloadkan pada IC mikrokontroler ATmega16. Adapun listing program dari bagian-bagian yang diuji adalah sebagai berikut:

a. Tampilan LCD

Pada program LCD dibagi menjadi 2, pertama yaitu pada saat timer belum berjalan atau countdown belum berjalan, program menampilkan nilai array dengan penekanan tombol *keypad* yang nantinya digunakan untuk perhitungan variabel lamanya solenoid terbuka (*lama_sole*) dan lamanya waktu countdown. Seperti pada gambar 26. Kedua yaitu pada saat tombol OK ditekan dan timer berjalan, maka LCD menampilkan waktu *countdown*.



Gambar 26. Tampilan Untuk Waktu Buka Solenoid



Gambar 27. Tampilan Pada Saat Countdown

Adapun programnya sebagai berikut.

```
void lcd()
{
    if (TIMSK==0x82)
    {
        switch(j)
        {
            case 0: lcd_clear();
                    sprintf(screen2,"Set solenoid opn");
                    lcd_gotoxy(0,0);
                    lcd_puts(screen2);
                    switch (i)
                    {
                        case 0 :    sprintf(screen1,"[      ]
miliscnd");
                                    lcd_gotoxy(0,1);
                                    lcd_puts(screen1);
                                    break;
                        case 1 :    sprintf(screen1,"[   %i]
miliscnd",key[1]);
                                    lcd_gotoxy(0,1);
                                    lcd_puts(screen1);
                                    break;
                        case 2 :    sprintf(screen1,"[  %i%i]
miliscnd",key[1],key[2]);
                                    lcd_gotoxy(0,1);
                                    lcd_puts(screen1);
                                    break;
                        case 3 :    sprintf(screen1,"[
%i%i%i] miliscnd",key[1],key[2],key[3]);
                                    lcd_gotoxy(0,1);
                                    lcd_puts(screen1);
                                    break;
                    }
                }
            }
    }
```

```

                                case 4 :
sprintf(screen1,"[%i%i%i%i]
miliscnd",key[1],key[2],key[3],key[4]);
                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                }
                                break;
case 1: lcd_clear();
        sprintf(screen2,"Countdwn to open");
        lcd_gotoxy(0,0);
        lcd_puts(screen2);/*

        k = lama_sole/1000;
        l = lama_sole%1000;
        m = l/100;
        n = l%100;
        v = n/10;
        w = n%10;

        lcd_gotoxy(0,0);
        lcd_putchar(k+0x30);
        lcd_putchar(m+0x30);
        lcd_putchar(v+0x30);
        lcd_putchar(w+0x30);          */

        switch (i)
        {   case 5 :sprintf(screen1,"[   :   :
]");

                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                case 6 :sprintf(screen1," [%i :   :
]",key[6]);

                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                case 7 :sprintf(screen1," [%i%i :
:   ] ",key[6],key[7]);

                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                case 8 :sprintf(screen1," [%i%i : %i
:   ] ",key[6],key[7],key[8]);

                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                case 9 :sprintf(screen1," [%i%i :
%i%i :   ] ",key[6],key[7],key[8],key[9]);

                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                case 10 :sprintf(screen1," [%i%i :
%i%i : %i ] ",key[6],key[7],key[8],key[9],key[10]);

                                lcd_gotoxy(0,1);

```

```

                                lcd_puts(screen1);
                                break;
                                case 11 :sprintf(screen1," [%i%i :
%i%i : %i%i] ",key[6],key[7],key[8],key[9],key[10],key[11]);
                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                }
                                angka1=jam;
                                angka2=menit;
                                angka3=detik;

                                }
                                }

if (TIMSK==0x86 && x!=0)      // tombol OK sudah ditekan
{
    unsigned char a,b,c,d,e,f,g,h,o,p,q,r;

    a = jam/10;
    b = jam%10;

    c = menit/10;
    d = menit%10;

    e = detik/10;
    f = detik%10;
    /*
    g = lama_sole/1000;
    h = lama_sole%1000;
    o = h/100;
    p = h%100;
    q = p/10;
    r = p%10;      */

    lcd_clear();
    sprintf(screen2,"Countdown to open");
    lcd_gotoxy(0,0);
    lcd_puts(screen2);
    lcd_gotoxy(1,1);
    lcd_putchar(a+0x30);
    lcd_putchar(b+0x30);
    lcd_gotoxy(4,1);
    lcd_putsf(":");
    lcd_gotoxy(6,1);
    lcd_putchar(c+0x30);
    lcd_putchar(d+0x30);
    lcd_gotoxy(9,1);
    lcd_putsf(":");
    lcd_gotoxy(11,1);
    lcd_putchar(e+0x30);
    lcd_putchar(f+0x30);

```

b. Keypad

Pada program *keypad* yaitu untuk mengisi variabel *lama_ole* dan variabel jam, menit dan detik dengan cara memanggil fungsi nilai. Fungsi nilai tersebut didalamnya terdapat perhitungan variabel *lama_ole* dan waktu countdown yaitu variabel jam menit dan detik dengan hasil angka array dari penekanan tombol *keypad*. Tombol ok disini berfungsi menjadi 2 bagian. Pertama untuk memasukkan hasil penekanan tombol *keypad* ke perhitungan variabel *lama_ole* yaitu keadaan dimana *j=0*. Kedua pada saat keadaan *j=1* yang fungsinya untuk memasukkan hasil penekanan tombol *keypad* ke perhitungan variabel countdown. Tombol reset fungsinya yaitu untuk membuat semua variabel menjadi 0 dan mematikan timer 1 dan timer 0. Adapun program pada *keypad* yaitu sebagai berikut.

```
void keypad()                // scanning keypad
{
    PORTC=0b00111111;
    delay_ms(15);
    if (PINC.3==0)           // tombol Reset
    {
        i=0;
        j=0;
        key[1]=0;
        key[2]=0;
        key[3]=0;
        key[4]=0;
        key[5]=0;
        key[6]=0;
        key[7]=0;
        key[8]=0;
        key[9]=0;
        key[10]=0;
        key[11]=0;
        jam=0;
        menit=0;
        detik=0;
        TIMSK=0x82;
        delay_ms(200); }
    if (PINC.2==0)
    {
        i++;
```

```

        key[i]=9;
        nilai_countdown();
        delay_ms(200); }
if (PINC.1==0)
{
    i++;
    key[i]=6;
    nilai_countdown();
    delay_ms(200); }
if (PINC.0==0)
{
    i++;
    key[i]=3;
    nilai_countdown();
    delay_ms(200); }

PORTC=0b01011111;
delay_ms(15);
if (PINC.3==0)
{
    i++;
    key[i]=0;
    nilai_countdown();
    delay_ms(200); }
if (PINC.2==0)
{
    i++;
    key[i]=8;
    nilai_countdown();
    delay_ms(200); }
if (PINC.1==0)
{
    i++;
    key[i]=5;
    nilai_countdown();
    delay_ms(200); }
if (PINC.0==0)
{
    i++;
    key[i]=2;
    nilai_countdown();
    delay_ms(200); }

PORTC=0b01101111;
delay_ms(15);
if (PINC.3==0)          // tombol OK
{
    switch (j)
    {
        case 0 :j=1;
                i=5;
                //break;
                goto countdown;
        case 1 :if (jam>0 || menit>0 || detik>0)
                {
                    TIMSK=0x86;
                    x=2;
                    delay_ms(50);
                    break;
                }
    }
    countdown:
}

```



```

if (PINC.2==0)
{
    i++;
    key[i]=7;
    nilai_countdown();
    delay_ms(200);}
if (PINC.1==0)
{
    i++;
    key[i]=4;
    nilai_countdown();
    delay_ms(200);}
if (PINC.0==0)
{
    i++;
    key[i]=1;
    nilai_countdown();
    delay_ms(200);}
}

```

c. Tombol Menu

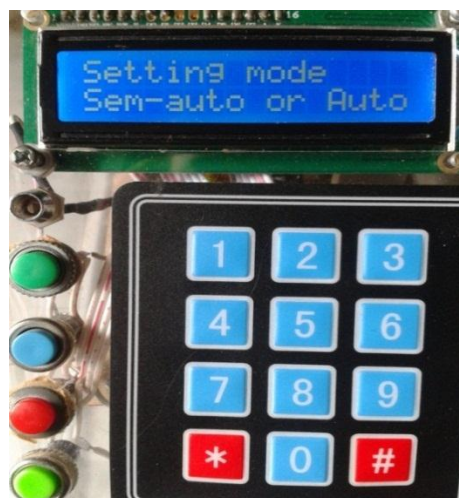
Program pada tombol menu:

```

if (menu==0)
{
    goto awal;
    TIMSK=0x82;
}

```

Dari program diatas dapat dijelaskan secara garis besar yaitu jika tombol menu ditekan maka program utama kembali ke menu pilihan, seperti pada gambar 28.



Gambar 28. Tampilan Untuk Tombol Menu

d. Tombol Manual

Tombol manual difungsikan untuk membuka solenoid secara manual. Jika tombol manual ditekan maka solenoid akan terbuka. Adapun listing programnya sebagai berikut.

```
if (man==0)          // tombol manual
{
    sole=1;
}
else
{
    sole=0; }

```

e. Tombol Semi-Auto

Pada tombol Semi-Auto pengguna hanya dapat melakukan kerja otomatis alat hanya dalam satu kali pengaturan waktu, setelah itu pengguna harus melakukan pen-*setting* ulang agar dapat melakukan kerja alat lagi. Seperti pada gambar 29.



Gambar 29. Tampilan Untuk Tombol Semi-Auto

Adapun listing programnya adalah sebagai berikut.

```

if (semoto==0)    // mode semi otomatis
{
    lcd_clear();
    delay_ms(800);
    lcd_gotoxy(0,0);
    lcd_putsf("Mode Semi-Auto");
    lcd_clear();
    delay_ms(600);
    lcd_gotoxy(0,0);
    lcd_putsf("Atur Waktu");
    delay_ms(800);
    lcd_gotoxy(0,0);
    lcd_putsf("  * => OK    ");
    lcd_gotoxy(0,1);
    lcd_putsf("  # => Reset  ");
    delay_ms(800);

    while(1)
    {
        if (menu==0)
        {
            goto awal;
            TIMSK=0x82;
        }
        keypad();
        lcd();
        adc();
        nilai_adc();

        if (jam==0 && menit==0 && detik==0 && x==1) //
keadaan countdown = 0
        {
            x=0;
            lcd_clear();
            lcd_gotoxy(3,0);
            lcd_putsf("Setting lagi");
            lcd_gotoxy(1,1);
            lcd_putsf("Tekan #");
            delay_ms(800);
        }
    }
}

```

f. Tombol Otomatis

Pada tombol otomatis pengguna dapat melakukan kerja otomatis alat dalam satu kali pengaturan waktu dan alat akan terus bekerja sebelum pengguna melakukan reset. Adapun listing programnya adalah sebagai berikut.

```

if (oto==0)          // mode full otomatis
{
    lcd_clear();
    delay_ms(800);
    lcd_gotoxy(0,0);
    lcd_putsf("Mde Full-Auto");
    lcd_clear();
    delay_ms(600);
    lcd_gotoxy(0,0);
    lcd_putsf("Atur Waktu");
    delay_ms(800);
    lcd_gotoxy(0,0);
    lcd_putsf("  * => OK      ");
    lcd_gotoxy(0,1);
    lcd_putsf("  # => Reset   ");
    delay_ms(850);

    while(1)
    {
        if (menu==0)
        {
            goto awal;
            TIMSK=0x82;
        }
        keypad();
        lcd();
        adc();
        nilai_adc();

        if (jam==0 && menit==0 && detik==0 && x==1) //
keadaan countdown = 0
        {
            x=0;
            delay_ms(50);
            jam=angka1;
            menit=angka2;
            detik=angka3;
            x=2;          // countdwn jalan lagi keadaan awal
        }
    }
}

```

g. Solenoid

Pada program ini menjelaskan solenoid akan terbuka jika variabel jam, menit dan detik sama dengan 0 dengan variabel lama_ole yang telah di setting dari awal. Adapun listing programnya sebagai berikut.

```

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{

```

```

// Reinitialize Timer1 value
TCNT1H=0xD5D0 >> 8;
TCNT1L=0xD5D0 & 0xff;
// Place your code here

if (x==2)
{

if (detik==0)
{
    if (menit==0)
    {
        if (menit==0 && jam==0 && detik==0)
        {
            sole=1;
            lcd_clear();
            lcd_gotoxy(5,0);
            lcd_putsf("open");
            //TCCR2=0x02;
            delay_ms(lama_sole);
            sole=0;
            x=1;
            i=0;
            j=0;
        }

        if (x!=1)
        { jam=jam-1;
          lcd();
          menit=60; }
    }

    if (x!=1)
    { menit=menit-1;
      lcd();
      detik=60; }
}

if (x!=1)
{ detik=detik-1;
  lcd(); }

}
}

```

h. LED Indikator

Pada program LED indikator menjelaskan jika nilai ADC lebih dari 900 maka timer0 aktif sehingga program *interrupt* dijalankan untuk

menghidupkan LED indikator pada alat. Dan program pada LED indikator adalah sebagai berikut.

```
void adc()
{
    if (read_adc(0)>900)
    {
        TCCR0=0x02;
    }
    else
    {
        TCCR0=0x00;
        idktr=0;
    }
}
```

4. Pengamatan Ketepatan Waktu Alat

Dalam pengujian kesesuaian waktu antara waktu yang sesungguhnya dengan waktu pada alat didapat hasil yang memuaskan dengan kesesuaian waktu sampai dengan detik. Ini dikarenakan menggunakan XTAL sebesar 11.059.200 Mhz, prescaler 1024, dan timer 1 overflow interrupt service routin. Kesesuaian waktu ini tidak lepas dari perhitungan nilai TCNT. Untuk mendapatkan waktu timer menjadi 1 detik, digunakan persamaan:

$$TCNT = (1 + FFFFh) - \left(\frac{T_{timer} \times f_{CLK}}{N} \right)$$

Dimana:

TCNT : Nilai timer (Hex)

f_{CLK} : Frekuensi clock (crystal) yang digunakan (Hz)

T_{timer} : Waktu timer yang diinginkan (detik)

N : Prescaler (1, 8, 64, 256, 1024)

1+FFFFh : Nilai maksimum timer adalah FFFFh dan overflow saat FFFFh ke 0000h

Pada program timer penulis menggunakan nilai heksadesimal sebesar D5D0. Nilai heksadesimal ini untuk mendapatkan nilai sebesar 1 detik. Jika dimasukkan dalam persamaan, maka:

$$\begin{aligned}
 TCNT &= (1 + FFFFh) - \left(\frac{T_{timer} \times f_{CLK}}{N} \right) \\
 &= (1 + FFFFh) - \left(\frac{1 \times 11059200}{1024} \right) \\
 &= 1000h - 108000d \\
 &= 1000h - 2A30h \\
 &= D5D0h
 \end{aligned}$$

Krena register TCNT merupakan register 16 bit, maka pada program terdapat register TCNT1H dimasukkan nilai heksadesimal sebesar D5 dan pada register TCNT1L dimasukkan nilai heksadesimal D0.

5. Pengamatan Berat Pakan Yang Dikeluarkan

Pengamatan dilakukan dengan mengukur berat pakan menggunakan timbangan digital, selanjutnya dengan menghitung berat rata-rata dan didapatkan selisih atau persentase kesalahannya. Hasil rata-rata berat pakan yang diamati dibedakan menjadi tiga bagian, yaitu untuk waktu buka solenoid 500 ms, 1000 ms dan 1500 ms.

a. Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 500 ms

Hasil rata-rata berat pakan dengan waktu buka solenoid 500 ms yaitu:

$$\frac{1,46 + 1,56 + 1,66 + 1,65 + 1,67}{5} = 1,6 \text{ gr}$$

Selanjutnya yaitu perhitungan persentase kesalahan dalam lima kali percobaan dengan waktu buka solenoid 500 ms.

$$\text{Dengan berat pakan } 1,46 := \frac{|1,46 - 1,6|}{1,6} \times 100\% = 8,75\%$$

$$\text{Dengan berat pakan } 1,56 := \frac{|1,56 - 1,6|}{1,6} \times 100\% = 2,5\%$$

$$\text{Dengan berat pakan } 1,66 := \frac{|1,66 - 1,6|}{1,6} \times 100\% = 3,75\%$$

$$\text{Dengan berat pakan } 1,65 := \frac{|1,65 - 1,6|}{1,6} \times 100\% = 3,1\%$$

$$\text{Dengan berat pakan } 1,67 := \frac{|1,67 - 1,6|}{1,6} \times 100\% = 4,37\%$$

Persentase kesalahan diatas yaitu 8,75%, 2,5%, 3,75%, 3,1% dan 4,37%. Dari hasil perhitungan tersebut didapatkan hasil rata-rata persentase kesalahan.

$$\text{Dengan waktu buka 500 ms} := \frac{8,75 + 2,5 + 3,75 + 3,1 + 4,37}{5} = 4,49\%$$

Dapat disimpulkan bahwa persentase kesalahan pada alat relatif kecil.

b. Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 1000 ms

Hasil rata-rata berat pakan dengan waktu buka solenoid 1000 ms yaitu:

$$\frac{4,84 + 4,74 + 4,83 + 4,81 + 4,86}{5} = 4,82 \text{ gr}$$

Selanjutnya yaitu perhitungan persentase kesalahan dalam lima kali percobaan dengan waktu buka solenoid 1000 ms.

$$\text{Dengan berat pakan } 4,84 := \frac{|4,84 - 4,82|}{4,82} \times 100\% = 0,41\%$$

$$\text{Dengan berat pakan } 4,74 := \frac{|4,74 - 4,82|}{4,82} \times 100\% = 1,66\%$$

$$\text{Dengan berat pakan } 4,83 := \frac{|4,83 - 4,82|}{4,82} \times 100\% = 0,21\%$$

$$\text{Dengan berat pakan } 4,81 := \frac{|4,81 - 4,82|}{4,82} \times 100\% = 0,21\%$$

$$\text{Dengan berat pakan } 4,86 := \frac{|4,86 - 4,82|}{4,82} \times 100\% = 0,83\%$$

Persentase kesalahan diatas yaitu 0,41%, 1,66%, 0,21%, 0,21% dan 0,83%. Dari hasil perhitungan tersebut didapatkan hasil rata-rata persentase kesalahan.

$$\text{Dengan waktu buka } 1000 \text{ ms} := \frac{0,41 + 1,66 + 0,21 + 0,21 + 0,83}{5} = 0,66\%$$

Dapat disimpulkan bahwa persentase kesalahan pada alat relatif kecil.

c. Berat Pakan Yang Dikeluarkan Dengan Waktu Buka Solenoid 1500 ms

Hasil rata-rata berat pakan dengan waktu buka solenoid 1500 ms yaitu:

$$\frac{8,28 + 8,42 + 8,37 + 8,36 + 8,32}{5} = 8,35 \text{ gr}$$

Selanjutnya yaitu perhitungan persentase kesalahan dalam lima kali percobaan dengan waktu buka solenoid 1500 ms.

$$\text{Dengan berat pakan } 8,28 := \frac{|8,28 - 8,35|}{8,35} \times 100\% = 0,84\%$$

$$\text{Dengan berat pakan } 8,42 := \frac{|8,42 - 8,35|}{8,35} \times 100\% = 0,84\%$$

$$\text{Dengan berat pakan } 8,37 := \frac{|8,37 - 8,35|}{8,35} \times 100\% = 0,24\%$$

$$\text{Dengan berat pakan } 8,36 := \frac{|8,36 - 8,35|}{8,35} \times 100\% = 0,12\%$$

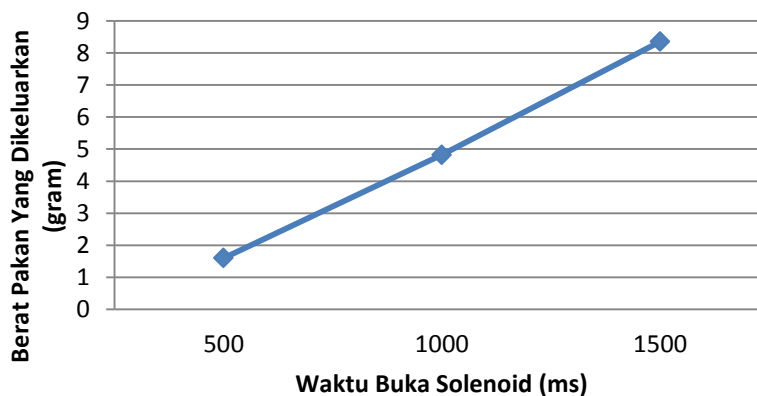
$$\text{Dengan berat pakan } 8,32 := \frac{|8,32 - 8,35|}{8,35} \times 100\% = 0,36\%$$

Persentase kesalahan diatas yaitu 0,84%, 0,84%, 0,24%, 0,12% dan 0,36%. Dari hasil perhitungan tersebut didapatkan hasil rata-rata persentase kesalahan.

$$\text{Dengan waktu buka } 1500 \text{ ms} := \frac{0,84 + 0,84 + 0,24 + 0,12 + 0,36}{5} = 0,48\%$$

Dapat disimpulkan bahwa persentase kesalahan pada alat relatif kecil.

Hasil perhitungan diatas dapat membedakan berat pakan dalam tiga jenis waktu buka pada solenoid. Pertama pada waktu buka solenoid 500 ms, berat pakan yang dikeluarkan rata-rata yaitu sebanyak 1,6 gram dengan persentase kesalahan rata-rata sebesar 4,49%. Kedua pada waktu buka solenoid 1000 ms, berat pakan yang dikeluarkan rata-rata yaitu sebanyak 4,82 gram dengan persentase kesalahan rata-rata sebesar 0,66%. Dan ketiga pada waktu buka solenoid 1500 ms, berat pakan yang dikeluarkan rata-rata yaitu sebanyak 8,35 gram dengan persentase kesalahan sebesar 0,48%. Dari hasil tersebut didapatkan grafik berat pakan yang dikeluarkan seperti pada gambar 30.



Gambar 30. Grafik Berat Pakan Yang Dikeluarkan

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan uraian perancangan, proses pembuatan dan pembahasan mengenai “Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16” maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Rancangan “Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16” memiliki empat bagian yaitu catu daya, sistem minimum, rangkaian driver dan program. Catu daya berfungsi sebagai penyuplai tegangan. Sistem *minimum* berupa rangkaian elektronik yang berfungsi sebagai pengolah data dengan mikrokontroler ATmega16 sebagai pusat kendali. Rangkaian driver yang berfungsi untuk mengatur buka tutup solenoid pada alat pemberi makan ikan. Dan program yang berfungsi untuk mengatur mikrokontroler sehingga alat dapat bekerja sesuai dengan fitur yang ditawarkan.
2. Untuk kerja dari “Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16” telah menunjukkan hasil yang sesuai dengan perencanaan yaitu alat dapat memberikan pakan ikan secara otomatis pada waktu yang telah ditentukan. Rata-rata berat pakan yang dikeluarkan yaitu sebesar 1,6 gram pada waktu buka solenoid 500 ms, 4,82

gram pada waktu buka solenoid 1000 ms dan 8,35 gram pada waktu buka solenoid 1500 ms .

B. Keterbatasan Alat

Keterbatasan “Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16” ini antara lain:

1. Alat ini tidak dilengkapi dengan *backup* catu daya sebagai catu daya cadangan, sehingga jika terjadi gangguan dari PLN seperti pemadaman listrik maka alat ini tidak akan bekerja sebagaimana mestinya.
2. Alat ini hanya dapat dipasang pada aquarium ukuran tertentu, sehingga diperlukan pengembangan mekanik agar alat dapat digunakan dengan berbagai jenis ukuran aquarium.
3. Berat pakan yang dikeluarkan dengan waktu buka solenoid yang berbeda masih belum linear berdasarkan hasil pengujian.

C. Saran

1. Pada “Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16” perlu ditambahkan *backup* catu daya.
2. Pada “Alat Pemberi Makan Ikan Di Aquarium Otomatis Berbasis Mikrokontroler ATmega16” perlu dilakukan pengembangan terhadap mekaniknya, sehingga alat ini dapat digunakan pada aquarium dengan berbagai jenis ukuran.
3. Perlu dilakukan pengembangan terhadap tempat pakan ikannya, sehingga berat pakan yang dikeluarkan menjadi linear.

DAFTAR PUSTAKA

- Afrie Setiawan. (2011). 20 Aplikasi Mikrokontroler ATmega16 Menggunakan BASCOM-AVR. Yogyakarta: ANDI
- Artanto, Dian , S.T., M.Eng. (2007). *Diklat kuliah Algoritma Pemrograman*. Yogyakarta: FST-USD
- Efvy Zanidra Zam. (2002). Mudah Menguasai Elektronika. Surabaya: Indah
- Feri Andang. (2007). *Box Packaging Controller Menggunakan Mikrokontroller AT89S51*. Yogyakarta: Universita Negeri Yogyakarta
- Heri Andriyanto. (2008). *Pemrograman Mikrokontroler AVR ATmega16*. Bandung: Informatika
- Sunomo. (1996). *Elektronika II*. Universita Negeri Yogyakarta
- Sunyoto. (1993). *Mesin Listrik Arus Searah*. Yogyakarta: Universita Negeri Yogyakarta
- Sutrisno. (1986). *Elektronika Teori Dasar dan Penerapannya*. Bandung: ITB
- Zaki M.H. (2008). *Cara Mudah Belajar Merangkai Elektronika Dasar*. Yogyakarta: AbsolutOK
- NN .____. *Datasheet BD135/137/139* Diakses 30 Januari 2013 pukul 22.00 WIB dari <http://www.fairchildsemi.com/ds/BD/BD135.pdf>
- NN .____. *Datasheet Pushbutton* Diakses 28 Januari 2013 pukul 23.00 WIB dari <http://shpat.com/docs/grayhill/pushbuttons.pdf>

NN .____. *Datasheet Resistor* Diakses 29 Januari 2013 pukul 23.30 WIB dari <http://www.vishay.com/docs/28729/28729.pdf>

Anonim. (2013). *Elektronika Dasar*. Diakses 30 Januari 2013 pukul 20.45 WIB dari <http://elektronika-dasar.web.id/komponen/led-light-light-emitting-dioda/>

Anonim. (2013). *Elektronik Tutorial*. Diakses 30 Januari 2013 pukul 21.45 WIB dari http://www.electronics-tutorials.ws/io/io_6.html

LAMPIRAN

Lampiran 1. Listing Program Lengkap

```

/*****
This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project : Alat Pemberi Makan Ikan Di Aquarium Otomatis
Version : V 0.1.
Date    : 2/4/2014
Author  : Lukman Nulhakim
Company : UNY
Comments:

Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 11.059200 MHz
Memory model        : Small
External RAM size    : 0
Data Stack size     : 128
*****/

#include <mega16.h>
#include <stdio.h>
#include <delay.h>

// Alphanumeric LCD functions
#include <alcd.h>

#define man      PINA.7
#define menu     PINA.6
#define semoto   PINA.5
#define oto      PINA.4
#define sole     PORTD.7
#define idktr    PORTA.3
#define cek_adc  PINC.7

unsigned char jam, menit, detik, j=0, angka1, angka2, angka3;
char screen1[16], screen2[16];
unsigned int i=0, m, lama_sole, lama1;
unsigned char x=0, key[12];

void nilai_countdown()
{
    if (i==1)
    { lama_sole=key[1]; }
    if (i==2)
    { lama_sole=key[1]*10+key[2]; }
    if (i==3)
    { lama_sole=key[1]*100+key[2]*10+key[3]; }
    if (i==4)
    { lama_sole=key[1]*1000+key[2]*100+key[3]*10+key[4]; }
}
```



```

        if (i==6)
        {
            jam=key[6];
        }
        if (i==7)
        {   jam=key[6]*10+key[7]; }
        if (i==8)
        {   if (key[i]>=6)
            {   key[i]=0;
                i=i-1;
                lcd_clear();
                lcd_gotoxy(2,0);
                lcd_putsf("maks 59 menit");
                delay_ms(1000);}
            menit=key[8];
        }
        if (i==9)
        {   menit=key[8]*10+key[9]; }
        if (i==10)
        {   if (key[i]>=6)
            {   key[i]=0;
                i=i-1;
                lcd_clear();
                lcd_gotoxy(2,0);
                lcd_putsf("maks 59 detik");
                delay_ms(1000);}
            detik=key[10]; }
        if (i==11)
        {   detik=key[10]*10+key[11]; }

    }

    bit set;
    unsigned int lama;
    // Timer 0 output compare interrupt service routine
    interrupt [TIM0_COMP] void timer0_comp_isr(void)
    {
        // Place your code here
        lama++;
        if (lama==2000)
        {
            if (set==0)
            {   idktr=1;
                set=1;
            }

            else if(set==1)
            {   idktr=0;
                set=0;
            }
            lama=0;
        }
    }

    // Timer 2 output compare interrupt service routine

```

```

interrupt [TIM2_COMP] void timer2_comp_isr(void)
{
    // Place your code here
    /*lama1++;
    if (lama1==lama_sole)
    {
        lcd_gotoxy(2,1);
        lcd_putsf("jadi");
        sole=0;
        delay_ms(200);
        x=1;
        i=0;
        j=0;
    } */
}

unsigned int k,l,m,n,v,w;
void lcd()
{
    if (TIMSK==0x82)
    {
        switch(j)
        {
            case 0: lcd_clear();
                    sprintf(screen2,"Set solenoid opn");
                    lcd_gotoxy(0,0);
                    lcd_puts(screen2);
                    switch (i)
                    {
                        case 0 :    sprintf(screen1,"[      ]
miliscnd");
                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                case 1 :    sprintf(screen1,"[   %i]
miliscnd",key[1]);
                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                case 2 :    sprintf(screen1,"[  %i%i]
miliscnd",key[1],key[2]);
                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                case 3 :    sprintf(screen1,"[ %i%i%i]
miliscnd",key[1],key[2],key[3]);
                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                                case 4 :    sprintf(screen1,"[%i%i%i%i]
miliscnd",key[1],key[2],key[3],key[4]);
                                lcd_gotoxy(0,1);
                                lcd_puts(screen1);
                                break;
                            }
                        break;
                    case 1: lcd_clear();
                            sprintf(screen2,"Countdown to open");

```

```

        lcd_gotoxy(0,0);
        lcd_puts(screen2);/*

k = lama_sole/1000;
l = lama_sole%1000;
m = l/100;
n = l%100;
v = n/10;
w = n%10;

        lcd_gotoxy(0,0);
        lcd_putchar(k+0x30);
        lcd_putchar(m+0x30);
        lcd_putchar(v+0x30);
        lcd_putchar(w+0x30);          */

switch (i)
{
    case 5 :sprintf(screen1,"[      :      :      ]");
            lcd_gotoxy(0,1);
            lcd_puts(screen1);
            break;
    case 6 :sprintf(screen1," [%i      :      :      ]",key[6]);
            lcd_gotoxy(0,1);
            lcd_puts(screen1);
            break;
    case 7 :sprintf(screen1," [%i%i      :      :      ]",key[6],key[7]);
            lcd_gotoxy(0,1);
            lcd_puts(screen1);
            break;
    case 8 :sprintf(screen1," [%i%i      : %i      :      ]",key[6],key[7],key[8]);
            lcd_gotoxy(0,1);
            lcd_puts(screen1);
            break;
    case 9 :sprintf(screen1," [%i%i      : %i%i      :      ]",key[6],key[7],key[8],key[9]);
            lcd_gotoxy(0,1);
            lcd_puts(screen1);
            break;
    case 10 :sprintf(screen1," [%i%i      : %i%i      : %i      ]",key[6],key[7],key[8],key[9],key[10]);
            lcd_gotoxy(0,1);
            lcd_puts(screen1);
            break;
    case 11 :sprintf(screen1," [%i%i      : %i%i      : %i%i      ]",key[6],key[7],key[8],key[9],key[10],key[11]);
            lcd_gotoxy(0,1);
            lcd_puts(screen1);
            break;
}
angka1=jam;
angka2=menit;
angka3=detik;

```

```

    }
}

if (TIMSK==0x86 && x!=0)    // tombol OK sudah ditekan
{
    unsigned char a,b,c,d,e,f,g,h,o,p,q,r;

    a = jam/10;
    b = jam%10;

    c = menit/10;
    d = menit%10;

    e = detik/10;
    f = detik%10;
    /*
    g = lama_sole/1000;
    h = lama_sole%1000;
    o = h/100;
    p = h%100;
    q = p/10;
    r = p%10;    */

    lcd_clear();
    sprintf(screen2,"Countdown to open");
    lcd_gotoxy(0,0);
    lcd_puts(screen2);
    lcd_gotoxy(1,1);
    lcd_putchar(a+0x30);
    lcd_putchar(b+0x30);
    lcd_gotoxy(4,1);
    lcd_putsf(":");
    lcd_gotoxy(6,1);
    lcd_putchar(c+0x30);
    lcd_putchar(d+0x30);
    lcd_gotoxy(9,1);
    lcd_putsf(":");
    lcd_gotoxy(11,1);
    lcd_putchar(e+0x30);
    lcd_putchar(f+0x30);
    /*
    lcd_gotoxy(0,0);
    lcd_putchar(g+0x30);
    lcd_putchar(o+0x30);
    lcd_putchar(q+0x30);
    lcd_putchar(r+0x30); */

    /*
    lcd_gotoxy(11,0);
    lcd_putchar(g+0x30);
    lcd_putchar(o+0x30);
    lcd_putchar(q+0x30);
    lcd_putchar(r+0x30); */
}

```

```

}

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    // Reinitialize Timer1 value
    TCNT1H=0xD5D0 >> 8;
    TCNT1L=0xD5D0 & 0xff;
    // Place your code here

    if (x==2)
    {

        if (detik==0)
        {
            if (menit==0)
            {
                if (menit==0 && jam==0 && detik==0)
                {
                    sole=1;
                    lcd_clear();
                    lcd_gotoxy(5,0);
                    lcd_putsf("open");
                    //TCCR2=0x02;
                    delay_ms(lama_sole);
                    sole=0;
                    x=1;
                    i=0;
                    j=0;
                }

                if (x!=1)
                { jam=jam-1;
                  lcd();
                  menit=60; }
            }

            if (x!=1)
            { menit=menit-1;
              lcd();
              detik=60; }
        }

        if (x!=1)
        { detik=detik-1;
          lcd(); }

    }
}

#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);

```

```

// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}

void keypad()          // scanning keypad
{
    PORTC=0b00111111;
    delay_ms(15);
    if (PINC.3==0)      // tombol Reset
    {
        i=0;
        j=0;
        key[1]=0;
        key[2]=0;
        key[3]=0;
        key[4]=0;
        key[5]=0;
        key[6]=0;
        key[7]=0;
        key[8]=0;
        key[9]=0;
        key[10]=0;
        key[11]=0;
        jam=0;
        menit=0;
        detik=0;
        TIMSK=0x82;
        delay_ms(200); }
    if (PINC.2==0)
    {
        i++;
        key[i]=9;
        nilai_countdown();
        delay_ms(200); }
    if (PINC.1==0)
    {
        i++;
        key[i]=6;
        nilai_countdown();
        delay_ms(200); }
    if (PINC.0==0)
    {
        i++;
        key[i]=3;
        nilai_countdown();
        delay_ms(200); }

    PORTC=0b01011111;
    delay_ms(15);
    if (PINC.3==0)
    {
        i++;
        key[i]=0;
        nilai_countdown();

```

```

        delay_ms(200);}
if (PINC.2==0)
{
    i++;
    key[i]=8;
    nilai_countdown();
    delay_ms(200);}
if (PINC.1==0)
{
    i++;
    key[i]=5;
    nilai_countdown();
    delay_ms(200);}
if (PINC.0==0)
{
    i++;
    key[i]=2;
    nilai_countdown();
    delay_ms(200);}

PORTC=0b01101111;
delay_ms(15);
if (PINC.3==0)          // tombol OK
{
    switch (j)
    {
        case 0 :j=1;
                i=5;
                //break;
                goto countdown;
        case 1 :if (jam>0 || menit>0 || detik>0)
                {
                    TIMSK=0x86;
                    x=2;
                    delay_ms(50);
                    break;
                }
    }
    countdown:
}
if (PINC.2==0)
{
    i++;
    key[i]=7;
    nilai_countdown();
    delay_ms(200);}
if (PINC.1==0)
{
    i++;
    key[i]=4;
    nilai_countdown();
    delay_ms(200);}
if (PINC.0==0)
{
    i++;
    key[i]=1;
    nilai_countdown();
    delay_ms(200);}
}

void adc()
{
    if (read_adc(0)>900)

```

```

    {
        TCCR0=0x02;
    }
    else
    {
        TCCR0=0x00;
        idktr=0;
    }
}

```

```

void nilai_adc()
{
    if (cek_adc==0)
    {
        char a,b,c,d,e,f;
        lcd_clear();
        a=read_adc(0)/1000;
        b=read_adc(0)%1000;
        c=b/100;
        d=b%100;
        e=d/10;
        f=d%10;
        lcd_gotoxy(5,1);
        lcd_putchar(0x30+a);
        lcd_putchar(0x30+c);
        lcd_putchar(0x30+e);
        lcd_putchar(0x30+f);
    }
}

```

// Declare your global variables here

```

void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTA=0xF4;
    DDRA=0x08;

    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTB=0x01;
    DDRB=0xFF;

    // Port C initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In

```



```

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x80;
DDRC=0x70;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x87;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 1382.400 kHz
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x96;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 10.800 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x05;
TCNT1H=0xD5;
TCNT1L=0xD0;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x96;

// External Interrupt(s) initialization

```

```

// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x82;

// USART initialization
// USART disabled
UCSRB=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 691.200 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: Free Running
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0xA4;
SFIOR&=0x0F;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTB Bit 7
// RD - PORTB Bit 6
// EN - PORTB Bit 5
// D4 - PORTB Bit 4
// D5 - PORTB Bit 3
// D6 - PORTB Bit 2
// D7 - PORTB Bit 1
// Characters/line: 16
lcd_init(16);

// Global enable interrupts
#asm("sei")

lcd_gotoxy(0,0);
lcd_putsf("    BISMILLAH    ");
delay_ms(500);
lcd_gotoxy(0,0);

```

```

lcd_putsf("Lukman N Hakim");
lcd_gotoxy(1,1);
lcd_putsf("09506131021");
delay_ms(1000);

awal:

lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Setting mode");
lcd_gotoxy(0,1);
lcd_putsf("Sem-auto or Auto");

while (1)
{
    // Place your code here
    adc();

    if (man==0)          // tombol manual
    {
        sole=1;
    }
    else
    {
        sole=0; }

    if (semoto==0)      // mode semi otomatis
    {
        lcd_clear();
        delay_ms(800);
        lcd_gotoxy(0,0);
        lcd_putsf("Mode Semi-Auto");
        lcd_clear();
        delay_ms(600);
        lcd_gotoxy(0,0);
        lcd_putsf("Atur Waktu");
        delay_ms(800);
        lcd_gotoxy(0,0);
        lcd_putsf(" * => OK      ");
        lcd_gotoxy(0,1);
        lcd_putsf(" # => Reset   ");
        delay_ms(800);

        while(1)
        {
            if (menu==0)
            {
                goto awal;
                TIMSK=0x82;
            }
            keypad();
            lcd();
            adc();
            nilai_adc();

            if (jam==0 && menit==0 && detik==0 && x==1) //
keadaan countdown = 0
            {
                x=0;
                lcd_clear();

```

```

        lcd_gotoxy(3,0);
        lcd_putsf("Setting lagi");
        lcd_gotoxy(1,1);
        lcd_putsf("Tekan #");
        delay_ms(800);
    }
}

if (oto==0)          // mode full otomatis
{
    lcd_clear();
    delay_ms(800);
    lcd_gotoxy(0,0);
    lcd_putsf("Mde Full-Auto");
    lcd_clear();
    delay_ms(600);
    lcd_gotoxy(0,0);
    lcd_putsf("Atur Waktu");
    delay_ms(800);
    lcd_gotoxy(0,0);
    lcd_putsf(" * => OK ");
    lcd_gotoxy(0,1);
    lcd_putsf(" # => Reset ");
    delay_ms(850);

    while(1)
    {
        if (menu==0)
        {
            goto awal;
            TIMSK=0x82;
        }
        keypad();
        lcd();
        adc();
        nilai_adc();

        if (jam==0 && menit==0 && detik==0 && x==1) //
keadaan countdown = 0
        {
            x=0;
            delay_ms(50);
            jam=angka1;
            menit=angka2;
            detik=angka3;
            x=2;          // countdwn jalan lagi

keadaan awal
        }
    }
}
}

```

Lampiran 2. Data Sheet Atmega16.

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 16 Kbytes of In-System Self-programmable Flash program memory
 - 512 Bytes EEPROM
 - 1 Kbyte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7V - 5.5V for ATmega16L
 - 4.5V - 5.5V for ATmega16
- Speed Grades
 - 0 - 8 MHz for ATmega16L
 - 0 - 16 MHz for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
 - Active: 1.1 mA
 - Idle Mode: 0.35 mA
 - Power-down Mode: < 1 µA



8-bit AVR[®]
Microcontroller
with 16K Bytes
In-System
Programmable
Flash

ATmega16
ATmega16L

Summary

Rev. 2466TS-AVR-07/10



Port B (PB7..PB0)	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega16 as listed on page 58.</p>
Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.</p> <p>Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 61.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega16 as listed on page 63.</p>
<u>RESET</u>	<p>Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 38. Shorter pulses are not guaranteed to generate a reset.</p>
XTAL1	<p>Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.</p>
XTAL2	<p>Output from the inverting Oscillator amplifier.</p>
AVCC	<p>AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.</p>
AREF	<p>AREF is the analog reference pin for the A/D Converter.</p>



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16 provides the following features: 16 Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1 Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC Digital supply voltage.

GND Ground.

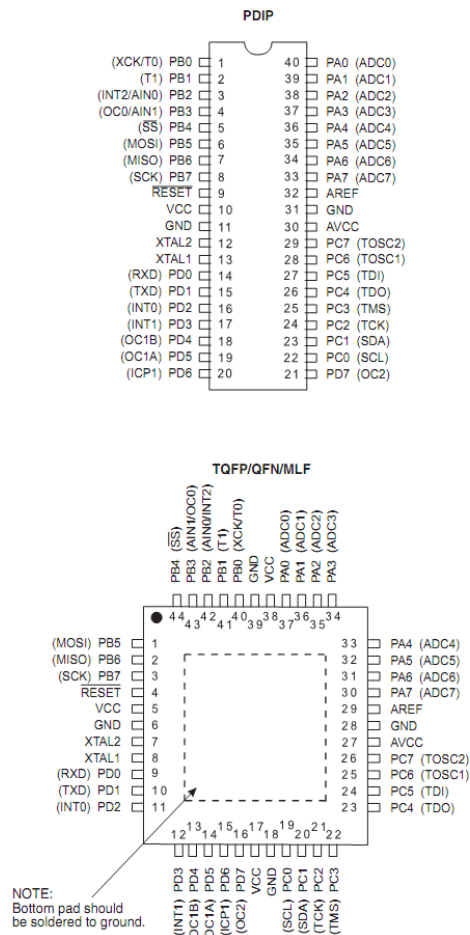
Port A (PA7..PA0) Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.



Pin Configurations

Figure 1. Pinout ATmega16



Disclaimer

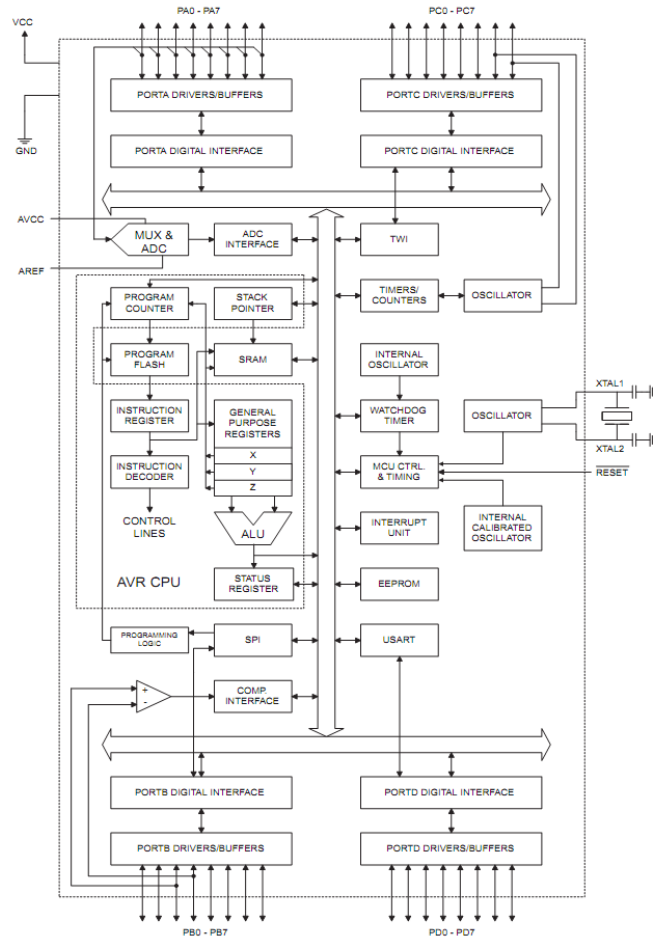
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 16 Kbytes of In-System Self-programmable Flash program memory
 - 512 Bytes EEPROM
 - 1 Kbyte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7V - 5.5V for ATmega16L
 - 4.5V - 5.5V for ATmega16
- Speed Grades
 - 0 - 8 MHz for ATmega16L
 - 0 - 16 MHz for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
 - Active: 1.1 mA
 - Idle Mode: 0.35 mA
 - Power-down Mode: < 1 µA



8-bit AVR[®]
Microcontroller
with 16K Bytes
In-System
Programmable
Flash


ATmega16
ATmega16L

Summary

Rev. 2466TS-AVR-07/10



Lampiran 3. Datasheet BD135/137/139



August 2013

BD135 / 137 / 139

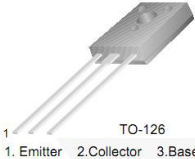
NPN Epitaxial Silicon Transistor

Features

- Complement to BD136, BD138 and BD140 respectively

Applications

- Medium Power Linear and Switching



TO-126
1. Emitter 2. Collector 3. Base

Ordering Information

Part Number	Marking	Package	Packing Method
BD13516S	BD135-16	TO-126 3L	Bulk
BD1356STU	BD135-6		Rail
BD13510STU	BD135-10		
BD13516STU	BD135-16		
BD13716STU	BD137-16		Bulk
BD13710STU	BD137-10		
BD13716S	BD137-16		Rail
BD13916STU	BD139-16		Bulk
BD13910S	BD139-10		
BD13916S	BD139-16		Rail
BD1396STU	BD139-6		
BD13910STU	BD139-10		

© 2007 Fairchild Semiconductor Corporation
BD135 / 137 / 139 Rev. 1.2.0

www.fairchildsemi.com

1

BD135 / 137 / 139 — NPN Epitaxial Silicon Transistor

Lampiran 4. Datasheet Pushbutton

Butt Contact Pushbutton Switches

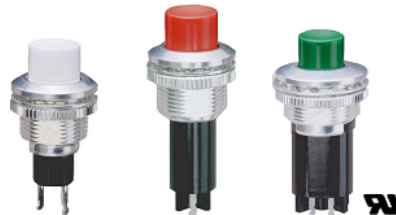


SERIES 30 and 46 Decorator Line

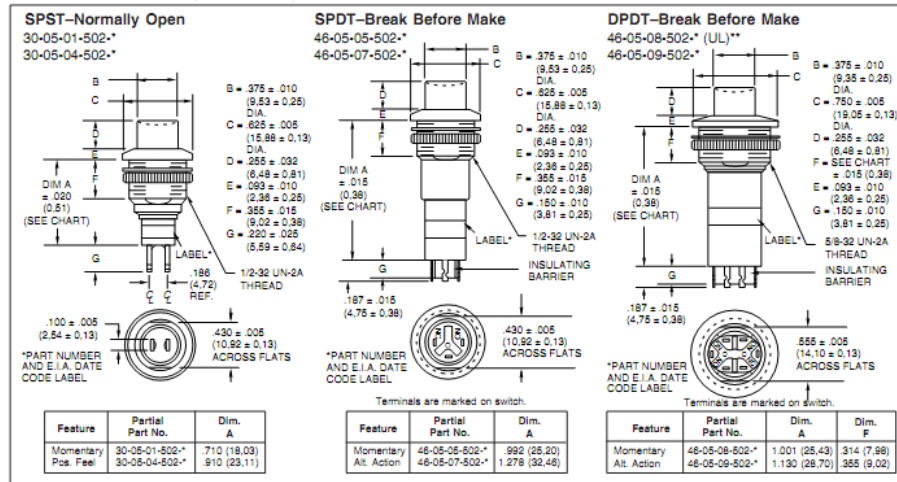


FEATURES

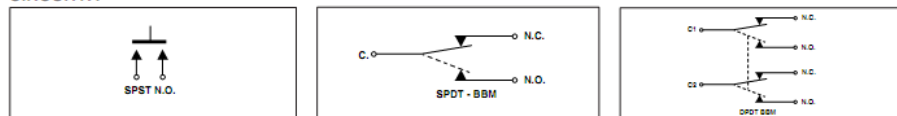
- Attractive Front Panel Bezels (6 choices)
- Space-Saving: Only 5/8" Diameter (SPST & SPDT) and 3/4" Diameter (DPDT)
- Round Bezels
- SPST with Option of Positive Feel
- SPDT, DPDT with Momentary Contact or Alternate Action
- Sturdy Metal Sleeve or Bushing and Nut Provide Secure Mounting
- UL Recognized Version



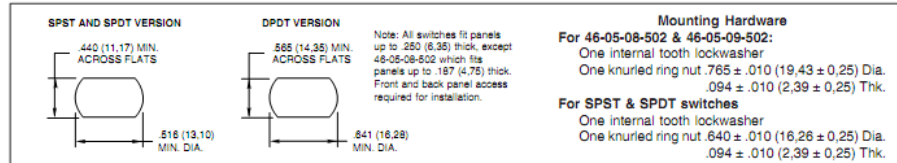
DIMENSIONS In Inches (and millimeters)



CIRCUITRY



RECOMMENDED PANEL CUTOUT AND MOUNTING HARDWARE



*Partial part number; complete part number by adding a dash for button color: white -01; red -03; green -04; blue -05; yellow -06 or black -07, see Ordering Information.
** Contact Grayhill for part number.

Lampiran 5. Datasheet Resistor

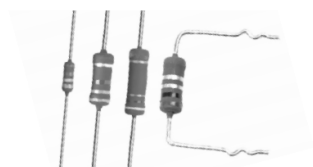


www.vishay.com

PR01/02/03

Vishay BCcomponents

Power Metal Film Leaded Resistors



DESCRIPTION

A homogeneous film of metal alloy is deposited on a high grade ceramic body. After a helical groove has been cut in the resistive layer, tinned connecting wires of electrolytic copper or copper-clad iron are welded to the end-caps. The resistors are coated with a red, non-flammable lacquer which provides electrical, mechanical and climatic protection. This coating is not resistant to aggressive fluxes and cleaning solvents. The encapsulation is resistant to all cleaning solvents in accordance with IEC 60068-2-45.

FEATURES

- High power in small packages (1 W/0207 size to 3 W/0617 size)
- Different lead materials for different applications
- Defined interruption behaviour
- Technology: Metal film
- AEC-Q200 qualified (PR01 and PR02)
- Lead (Pb)-free solder contacts
- Pure tin plating provides compatibility with lead (Pb)-free and lead containing soldering processes
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912



RoHS
COMPLIANT

APPLICATIONS

- All general purpose power applications

TECHNICAL SPECIFICATIONS						
DESCRIPTION	UNIT	PR01	PR02 Cu-lead	PR02 FeCu-lead	PR03 Cu-lead	PR03 FeCu-lead
Resistance range ⁽²⁾	Ω	0.22 to 1M	0.33 to 1M	1 to 1M	0.68 to 1M	1 to 1M
Resistance tolerance	%	$\pm 1; \pm 5$	$\pm 1; \pm 5$	$\pm 1; \pm 5$	$\pm 1; \pm 5$	$\pm 1; \pm 5$
Resistance series		± 1 (E24, E96); ± 5 (E24 series) ⁽¹⁾				
Rated dissipation, P_{70} $1 \Omega \leq R$	W	1	2	1.3	3	2.5
$R < 1 \Omega$		0.6	1.2	-	1.6	-
Thermal resistance ($R_{\theta J}$)	K/W	135	75	115	60	75
Temperature coefficient	ppm/K	$\leq \pm 250$	$\leq \pm 250$	$\leq \pm 250$	$\leq \pm 250$	$\leq \pm 250$
Maximum permissible voltage (U_{max} , AC/DC)	V	350	500	500	750	750
Basic specifications		IEC 60115-1				
Climatic category (IEC 60068-1)		55/155/56				
Stability after:						
Load (1000 h, P_{70})		ΔR max.: $\pm (5 \% R + 0.1 \Omega)$				
Long term damp heat test (56 days)		ΔR max.: $\pm (3 \% R + 0.1 \Omega)$				
Soldering (10 s, 260 °C)		ΔR max.: $\pm (1 \% R + 0.05 \Omega)$				

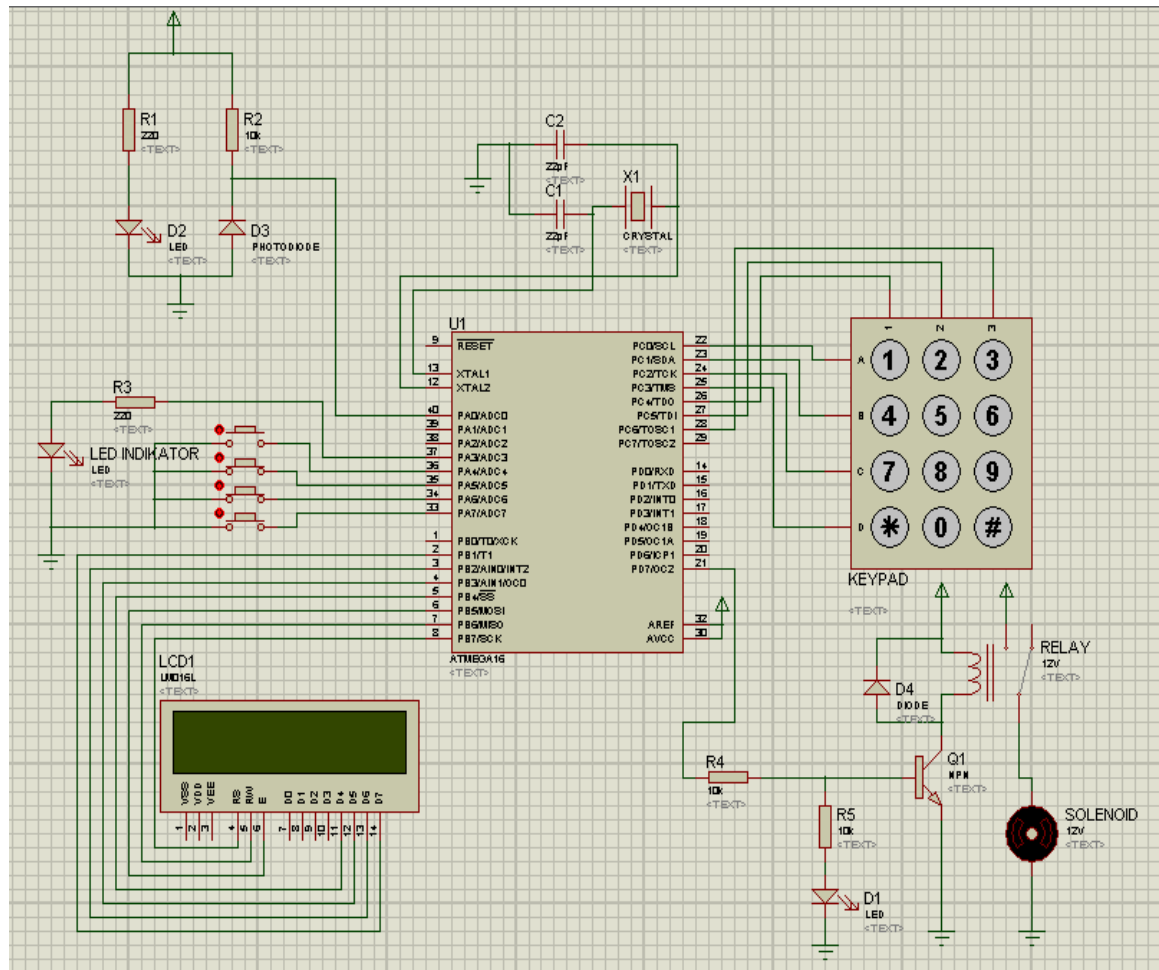
Notes

- R value is measured with probe distance of 24 mm \pm 1 mm using 4-terminal method.

⁽¹⁾ 1 % tolerance is available for R_n -range from 1 R upwards.

⁽²⁾ Ohmic values (other than resistance range) are available on request.

Lampiran 6. Rangkaian Keseluruhan



Lampiran 7. Foto Alat

